

المؤسسة الجامعية للدراسات والنشر والتوزيع - جدة



جلك ريفير

البرمجة : لغة المؤول (الاسمبلر)

ترجمة د. عبد الصن الحسني



البرمجة بلغة المؤول
(الـ اسمبلي)

جميع الحقوق محفوظة
الطبعة الأولى
1410 هـ - 1990 م

مكتبة **المعهد العلمي للبنان - بيروت**

بيروت - الطبع في بيروت - سنة 1410 هـ
مطبع: A-11111 - A-11111 - A-11111
بيروت - الطبعة - سنة 1410 هـ
م - 1410 هـ - 1410 هـ

سلسلة بإشراف
د. عبد الحسن الحسيني

جاك ريفيير

البرمجة : لغة المؤول (الاسمبلر)

ترجمة د. عبد الحسن الحسيني

الهيئة العامة للطباعة والنشر والتوزيع

هذا الكتاب ترجمة :

**LA PROGRAMMATION
EN ASSEMBLEUR**

Par

Jacques RIVIERE

تقديم

تعتبر لغة أسيمبلر (المؤول) من اللغات الفعالة وذات الإمكانيات الكبيرة نظراً لأنها تسمح للمبرمج باستعمال جميع إمكانيات ومقدرات وموارد الحاسب ، كما تسمح له بالدخول إلى « قلب » الآلة والعمل بالمرافص الداخلية للحاسب ، مما يضفي على البرنامج المكتوب بهذه اللغة فعالية كبيرة خصوصاً فيما يتعلق بالدقة والسرعة والعمل في الوقت الفعلي (real time) المستعمل كثيراً لإدارة العمليات الصناعية .

هذا الكتاب يُعالج لغة أسيمبلر الخاصة بعائلة الحاسبات IBM 360/370 التي شهدت إنتشاراً واسعاً في حقل المعلوماتية وأحدثت ثورة في صناعة الحاسبات في السنوات الأخيرة وبقيت تركيبة وهيكلية هذه الآلات مُستعملة وصالحة في وقتنا هذا وجرى إستعمالها والإفادة منها حتى في صناعة المعالج الصغري وتصميم الميكروحاسبات .

وبالنسبة للبرمجة بلغة المؤول ، فإن تقنية هذه البرمجة لا تختلف أبداً من آلة إلى أخرى ، صغيرة كانت أم كبيرة ، معالجاً صغرياً أو نظاماً كبيراً . أما الفرق الوحيد فيمكن في كون كود - الآلة يختلف من آلة إلى أخرى ، أما طريقة العمل والمعالجة واستعمال المرافص والذاكرة فلا تختلف إلا في عدد المرافص المبلوغة من المبرمج ، وبالتالي فإن التصرف على أي مؤول يبقى صالحاً بالنسبة لمعالج آخر بمؤول آخر .

وهنا يجب الإشارة إلى أن مؤول IBM/370 يتألف من أكبر عدد ممكن من التعليقات، وعدد مرافص الحاسب يعادل 16 للمعطيات و16 للعاوين ويستعمل عدداً كبيراً من طرق العنونة ، يصلح قسم منها لعنونة المعلومات عند إستعمال المعالج الصغري .

المترجم

تمهيد

لماذا كتاب جديد يختص بلغة المؤول (Assembler) ؟ وما هو المؤول ؟ هل تعرفون مُبرمجين يعملون بلغة المؤول حتى الآن ، بينما تقدّم اللغات المتطورة إمكانيات وتسهيلات جديدة ؟

كثيراً ما نسمع جميع هذه الأسئلة إضافة إلى أخرى مذهشة ، ولن نحاول هنا في هذا التمهيد أن نجاب عنها ، السؤال بعد الآخر ، ولكن سنحاول توضيح هدفنا من هذا الكتاب .

وُضع هذا الكتاب بسبب ثلاث ملاحظات :

- إن إتقان لغة المؤول هو الطريقة الأفضل لفهم طريقة عمل الحاسب .
- بواسطة إتقان لغة المؤول ، مهما يكن ، سنستطيع التفكير بسهولة أكثر وإدراك ماذا يحدث عندما نعمل بلغة أكثر تطوراً ، والبحث عن الأخطاء سيكون أكثر سهولة .
- عند نزول الميكروبروسسور إلى الأسواق ، أليس من الأفضل إتقان هذه اللغة الموجودة على هذه الآلات الصغيرة ؟ مع الإشارة إلى أن المؤول يبقى الوسيلة الفضلى لإنشاء وخلق المناهج الجديدة .

هكذا فلكتابنا هذا هدف تربوي . وهو ليس عبارة عن كتاب مساعد ومرجع في المعنى الذي نفهمه من المرجع المساعد الخاص بالنتج ، ولكنه عبارة عن مساعد كافٍ وكامل لفهم عمليات الإنشاء والبرمجة المهمة .

وهو موجّه إلى أولئك الراغبين بفهم طريقة عمل الآلات التي يستعملونها . ولقد حاولنا الإجابة عن المسائل التي ستواجهنا ، وبشكل خاص لدى الطلاب الذين يرغبون بمعرفة لغة المؤول بعد معرفتهم بإحدى اللغات المتطورة . وهذا هو دور الفصل الأول من الكتاب الذي يحتوي على عرض لتركيب وطريقة عمل الحاسب ، وهذا العرض جرى من خلال تفكير بسيط يتعلق بآلة ذات استعمال كبير : الحاسب الجيبى . ولأجل هؤلاء

أيضاً قمنا بعرض مشاكل العنونة ، التقطيع ، تنقيح الأربطة (link editor) ، الشحن (loading) ، والإنقطاعات عند الإدخال والإخراج (I/O interruption) . وهو مرجع أيضاً الى كل من يرغب بالعمل بلغة المؤول ، إما على الآلة المعتمدة كمرجع وهي الحاسب IBM 370 ، أو على الحاسب الشخصي الميكروكومبيوتر . وهنا نؤكد بأن جميع لغات التأويل هي متشابهة بشكل نستطيع معه بعد معرفة مؤول معين أن نتكيف بسهولة للعمل على مؤول آخر بآلة أخرى ، ولهذا الهدف قمنا بإضافة مسائل بسيطة ، تخدم التطبيق العملي لها على أغلب الحاسبات . وفي النهاية ، لهؤلاء الذين يعرفون المؤول ، قمنا بإثبات الإمكانات التي يقدمها التأويل المشروط وإستعمال الماكرو تعليمات (MACRO INSTRUCTIONS) . ونصائح هذا الكتاب التي تدور حول البرمجة الجيدة هي عبارة عن عناصر للتفكير يصبح في نهايتها البرنامج مختلفاً عن تلك المجموعة من التعليمات المبهمة كما في اللغة الثنائية . ومن الممكن إنشاء وتركيب برنامج مكتوب بلغة المؤول بشكل يصبح معه واضحاً كوضوح برنامج بلغة كوبول .

لماذا جرى إختيار الحاسب IBM 370 ؟

- لأنها شاملة وعامة . وأكثر صيغ لغة المؤول العاملة عليها جرى إستعمالها وتطويرها من قبل جميع المنتجين والصانعين .
- لماضيها ومستقبلها : إن المواصفات الخاصة بهذه اللغة والتي جهزت مع النظام IBM 360 ، قد جرت المحافظة عليها في الحاسبات IBM 370 وفي الأنظمة الجديدة من السلسلة 3000 و4000 إضافة إلى أغلب حاسبات IBM الجديدة .

القسم الأول

عموميات

1. آلة البسيطة

هذا الفصل الأول هو مخصص للمبتدئين . أما الذي يتمتع بمفاهيم كافية تتعلّق بهيكل المكتبة فيمكنه أن يبدأ دراسته من الفصل الثاني . إلّا أننا نعتقد بأنه يعرض ويوضح النقاط الأساسية لعملية الفهم اللاحقة . وهو يعرف المصطلحات الأساسية المتعلقة بدورة تنفيذ تعليمات الآلة .

1.1 . دراسة للآلة الحاسبة الصغيرة الجيبية

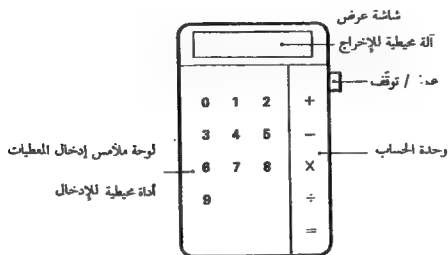
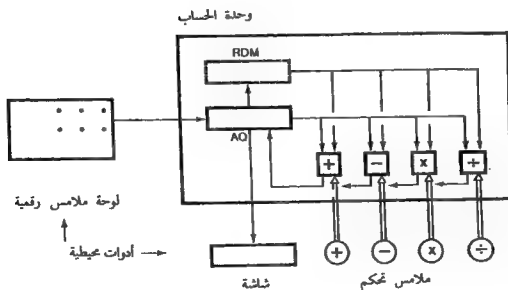
منذ النظرة الأولى ، تبدو الآلة الحاسبة الجيبية وكأنها مؤلفة من العناصر التالية :

- زر للعمل / ولوقف العمل .
- لوحة ملامس رقمية .
- شاشة للعرض .
- مجموعة من ملامس التحكم + ، - ، = ،
- فلنضم بعملية حساب بسيطة ، القسمة مثلاً . عملية المعالجة متجري كما يلي :
- 1 - وضع الآلة الحاسبة في العمل .
- 2 - إدخال العدد الأول (المقسوم) وعرضه .
- 3 - ضغط الزر الخاص بالقسمة .
- 4 - إدخال العدد الثاني (القاسم) وعرضه .
- 5 - الضغط على الزر = ، وعرض النتيجة .
- 6 - إيقاف عمل الآلة الحاسبة .

هذه السلسلة من العمليات تتطلب بعض الملاحظات :

- ترتيب العمليات هو مُحدّد وثابت ؛ لا يمكن عكس العمليات 2 و 4 .
- تتمتع مكتتنا ، إضافة إلى الدالة حساب (Compute) ، بدالة (مهمة) لإدخال المعطيات وبدالة لإخراج المعطيات (العرض على الشاشة) .
- عند إجراء العملية رقم 4 ، يختفي العدد المعروض على الشاشة ، قبل أن تتم عملية القسمة (يجب أن نعطي الصلاحية للعملية بالضغط على الملمس =) ، يجب إذاً ،

ويشكل إلزامي ، أن تحتوي المكنة على ذاكرة يُحْرَن فيها العدد الأول بانتظار نهاية إدخال القاسم . فلنعرض المخطط التوضيحي ⁽¹⁾ :

1.1 ~~1.1.2~~

2.1 مبحث

(1) إنَّ المخططات المعروضة في هذا الفصل لا تدعي تمثيل الدقة التكنولوجية ولكنها تعرض فقط الدلائل الأساسية القيدة للمبرمج .

هذا المخطط يُميّز بين نوعين من الخطوط . الخطوط البسيطة (→) والتي تناسب خطوط إنتقال المعطيات والخطوط المزدوجة (⇒) والتي تناسب خطوط تنقل الأوامر .

تعريفات :

نسَمي وحدة حساب مجموعة دارات الجمع والطرح ، ... مُخزّن معطيات الحساب في المناطق RDM وAQ والتي تدعى مرصيف (register) . المرصيف RDM يُستخدم لتخزين العدد الأول الداخِل إلى AQ للسّاح بإدخال العدد الثاني .

نتيجة الحساب توضع دائماً في مرصيف خاص AQ ولذلك نطلق عليه إسم مركم (Accumulator) . أمّا لوحة الملامس الرقمية وشاشة العرض فنطلق عليها الإسم : الأدوات المحيطة للإدخال والإخراج (I/O peripherals) .

2.1 . دراسة حاسبة جيبية مع ذاكرة

لننصف إلى الحاسبة الجيبية مجموعة من خلايا الذاكرة التي منطلق عليها الإسم : ذاكرة مركّزة (Central memory) . كل خلية من الذاكرة ، وتدعى أيضاً كلمة - آية (machine word) ، يمكنها كالمراصيف أن تحتوي على مخططات أو على نتائج الحساب . إلى كل خلية سريط عدداً معلداً يُدعى عنوان الخلية ويسمح بتمييز الخلايا فيما بينها . المؤثرات الأساسية (+ ، - ، ...) هي عبارة عن مؤثرات ثنائية (نقصد بذلك أنها تجري بين متأثرين (operators)) . أحد المتأثرين يكون موجوداً في المرصيف AQ والآخر في المرصيف RDM (مرصيف معطيات الذاكرة) . كما في الحاسبات البسيطة فإن النتيجة ستكون موجودة في AQ . يصبح من الضروري أن يكون بتصرفنا :

- نظام لإختيار العنوان الذي يؤمن الإتصال بين إحدى خلايا الذاكرة والمرصيف RDM ؛

- دارتان إضافيتان للشحن والترتيب ، لشحن مضمون خلية من الذاكرة في المركم وترتيب مضمون المركم في عنوان معيّن . هكذا دارات هي موجودة على جميع الحاسبات الجيبية وتتمتع بخلية ذاكرة واحدة على الأقل . مخطط حاسبة كهذه هو ممثّل على الشكل 3.1 .

إنّ منقاة العنوان هي هنا موضحة بواسطة ملامس دائري يؤمن الإتصال بين خلية من الذاكرة بعنوان معيّن ومضمون المرصيف RDM . ويتعلّق إنجاء إنتقال المعطيات بالمؤثر أو بالإشارة الحاسوبية المعتملة .

0	5	1	2
1	2	3	
2			

مثال حول عملية حساب بسيطة .
لنفترض، إن الذاكرة تحتوي على المعطيات التالية :

نرغب بجمع مضمون الخلية ذات العنوان 0 مع مضمون الخلية ذات العنوان 1 وبوضع النتيجة في العنوان 2 . فلنستعمل الترميز الكلاسيكي : (ALPHA) ، حيث ALPHA هي عبارة عن عنوان ، يشير الى مضمون الخلية ذات العنوان ALPHA . هكذا فإن (0) يعني هنا القيمة 125. السهم سيؤدي اتجاه انتقال المعطيات : AQ→(0) يعني خزن مضمون الخلية ذات العنوان (0) في المرمز AQ ، أي نخزن العدد 125 في AQ .

لإجراء عملية الحساب يجب :

- 1- تركيز منقاة العنوان على 0 والضغط على الزر «إشحن» ، مما يؤدي إلى تنفيذ العملية : AQ → (0) .
 - 2- تركيز منقاة العنوان على 1 والضغط على الزر + .
- هذا يسمح بإجراء العملية $AQ + (1) \rightarrow AQ$. هكذا فإن هذه العملية يمكن تقسيمها إلى إثنين .

$$(1) \rightarrow RDM$$

$$AQ + RDM \rightarrow AQ$$

- 3- تركيز منقاة العنوان على 2 والضغط على الزر «خزن» . هذا ما يسمح بتنفيذ العملية (2) : $AQ \rightarrow$.
- في نهاية هذه العمليات ، تحتوي الخلية ذات العنوان 2 على العدد 157 . والمرصف AQ يحتوي على القيمة النهائية .

ملاحظات :

جميع عمليات الحساب تتم بين المراسف AQ وRDM وليس من الذاكرة إلى الذاكرة . وهذا ما يؤدي إلى الحاجة إلى إجراء عملية شحن مسبقة للمرمز . المراسف هي إنفاً عبارة عن ذاكرة مرتبطة مباشرة بدارات الحساب .

للإشارة إلى مضمون خلايا الذاكرة سنستخدم على الترميز (عنوان adresse) بشكل نستطيع معه تمييز العنوان عن مضمونه ، أي إسم «nom» الخلية وقيمتها . المراسف المذكورة لا ترد داخل أهلة لأنه لا يوجد أي خلط ممكن بين المضمون والإسم : نعود دائماً إلى مضمون المراسف .

- 3.1 . من الحاسبة الصغيرة إلى الحاسب الكبير (الكومبيوتر)
إن كل معالجة تتناول معطيات وتسلسلاً دقيقاً من الأفعال ، والأوامر على الملامس + ، - ، ... ونوع الحاسبة المعتمدة حتى الآن لا يسمح بتخزين معطيات المسألة .

الفرق الأكبر بين الحاسبة ذات الذاكرة والحاسب الكبير يكمن في كون الأخير :
 - يُجَزَّن ليس فقط المعطيات ولكن الأوامر المطلوب إجراؤها على المعطيات .
 - يتمتع بأولية لربط الأوامر التي ستسمح له بتنفيذ هذه الأوامر حسب الترتيب الواردة فيه . هكذا ، فذاكرة الحاسب المركزية (C.M) تحتوي على معطيات المسألة وطريقة معالجتها للحصول على النتائج .

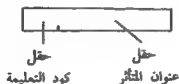
تعريفات :

في البداية ، سنعي كلمة أمر (Command) بالتعليمية (instruction) أو التعليمية الآلية (machine instruction) . ومجموعة التعليمات والمعطيات المرتبطة بها تُولف البرنامج . أما الملامس + ، - فستختفي . ويصبح عندئذ من البديهي أن لا يعمل الحاسب إلا إذا كان البرنامج مسجلاً في ذاكرته المركزية .

1.3.1 - هيكلية التعليمات الآلية

حسب المثل المذكور أعلاه في الفقرة 2.1 ، نستطيع أن نقول أن التعليمات الآلية هي مؤلفة من معلومتين :
 1- رقم يدل على الدارة المعتمدة من الوحدة المركزية .
 2- رقم يدل على عنوان المتأثر (Operand) .

إذا كانت التعليمية تعمل بمتأثرين (الحالة + ، - ، . . .) ، يكون المتأثر الأول مشحوناً مسبقاً في المركم (ACC) . هاتان المعلومتان ستكونان موجودتين في كلمة من الذاكرة بشكل مكود رقمياً ، مثلاً حسب الطريقة التالية :



وستسمح أوالية تكويد التعليمية ، التي سنقوم بتوضيحها لاحقاً ، بكشف ومعرفة الفعل المطلوب إجراؤه على المتأثر الموجود على العنوان المذكور في التعليمية .
 مثلاً :

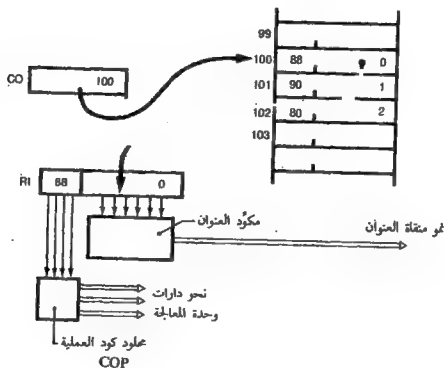
لفترض بأن كود عملية الشحن COP هو 88 ، وإن كود الجمع هو 90 وكود الخزن هو 80 . فلنخزن البرنامج الذي يقوم بجمع الخليتين 0 و 1 مع وضع النتيجة على العنوان 2 ، بدءاً من العنوان 100 . نحصل عندئذ على صورة الذاكرة التالية :

99		
100	8 8	0
101	9 0	1
102	8 0	2
103		

تنفيذ البرنامج يفترض ربطاً متتالياً للتعليمات الموجودة ، بدءاً من العنوان 100 ثم 101 ، ...

2.3.1 . أولية معرفة وربط التعليمات

تحتوي الذاكرة على نوعين من المعلومات بطبيعة دلالية مختلفة . المعطيات والتعليمات . من الضروري معانية ومعرفة الخلية التي تحتوي على التعليمة المطلوب تنفيذها . لهذا الهدف ، هناك مرصف خاص يسمى العدّاد الرئيسي الترتيبي (CO) أو عدّاد البرنامج program counter الذي سيحتوي في كل لحظة على العنوان التالي للتعليمة المطلوب تنفيذها . وبشكل خاص ، وفي البداية ، سيكون مشحوناً بعنوان أول تعليمة .



4.1 خطط

منذ اللحظة التي يجتري فيها CO على عنوان التعليم ، فإن دورة التنفيذ تبدأ :

1 - إرسال التعليم التي يشير إليها عداد البرنامج إلى مرصف التعليم RI المرتبط بمكوّد للعملية COP ومغقة العنوان .

2 - تكوين العنوان الذي يقوم بتركيز مغقة العنوان ، ومحمود (يفك كود) COP الذي يضع الدارة المناسبة من وحدة المعالجة في حالة العمل .

3 - تنفيذ العملية المطلوبة بواسطة وحدة المعالجة التي تستصح في طور العمل .

خلال المرحلة الثانية لن يكون من الضروري أن يؤثر CO على التعليم الموجودة في طور التنفيذ ، وخلال هذه المرحلة إذا تزداد قيمة عداد البرنامج CO واحداً (1) ليؤثر على التعليم التالية المطلوب تنفيذها .

بعد تنفيذ التعليم ، يعود الحاسب الى المرحلة الأولى بالقيمة الجديدة لعداد البرنامج CO وهذا يتتابع حتى نلتقي تعليم خاصة بوقف البرنامج .

يقي أن نشر إلى مختلف مراحل التنفيذ هي متزامنة بواسطة نبضات ساعة داخلية .

المخطط 5.1 التالي يعرض لمختلف المهام التي درستها . وهو يشكل المخطط العملي للحاسب .

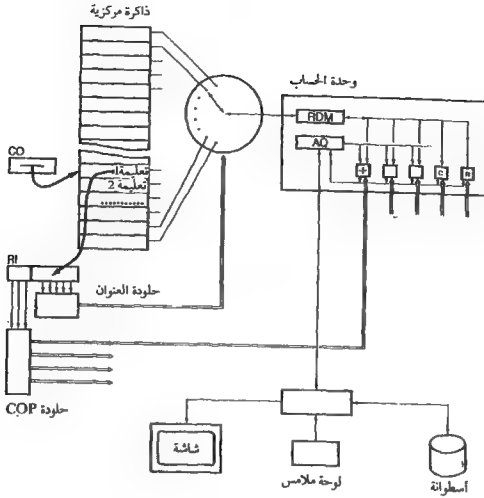
4.1 - خلاصة حول المكنة البسيطة

سنقوم بتوضيح الصيغ العملية للحاسب . إن جميع المكنات تستعمل هذه الأليات الأساسية ، إضافة إلى بعض التعديلات التي سندرسها عند الحاجة . فلنحاول الآن أن نستخلص بعض الملاحظات .

ملاحظة 1

المكنة المشروحة أعلاه هي مكنة « بعنوان بسيط » ، أي أن التعليم الآلية لا تراجع سوى عنوان واحد وإذن متأثر واحد علني . في هذه الحالة ، لنفترض عدداً كبيراً من المؤثرات (operators) تستعمل متأثرين والنتيجة ، ذلك يعني أن أحد المتأثرين ثم النتيجة موجودان في المرم . على بعض المكنات الأخرى قد نجد تعليمات تدعى « بعنوان مزدوج » .

(1) عندما تكون التعليمات ذات أطوال متغيرة (حالة الحاسبات IBM 360/370) يعقّم العدّاد C'D بمقدار طول التعليم .



مخطط 5.1 - الحاسب ، المخطط العملي

ملاحظة 2

لا تحتوي مكتبتنا سوى مركب واحد . هناك حاسبات أكثر فعالية يمكن أن تحتوي على عدد من المرافف التي تلعب دور المركب (هذه هي حالة المكنة IBM 360/370) . سيكون من الضروري أن نشير ، من داخل التعليمات ، إلى رقم المرفف الذي نعتنقه كمركب .

ملاحظة 3

نفترض ، كما في المخطط 3.1 ، أن ذاكرة المكنة تحتوي على 1000 خلية مرقمة من 0 إلى 999 . وهذا يعني أنّ :

- 1 - عداد البرنامج يحتوي على الأقل على ثلاثة مواقع عشرية تسمح له بمراجعة جميع عناوين الذاكرة المركزية ؛

2- أن حقل عنوان التعليم ، ولنفس السبب ، يجب أن يسمح بتسجيل الأعداد من 0 إلى 999 .

ملاحظة 4

بعض التعليقات يمكن أن لا تُراجع بواسطة عنوان ما . تظهر هذه الحالة ، مثلاً ، عندما لا نستعمل سوى AQ (عكس إشارة AQ ، تفسير AQ ، الإزاحة ، . . .) . ولكن من الممكن ، عند الحاجة ، إستعمال حقل العنوان لغايات أخرى . قد يحدث ، على بعض المكتات ، أن يكون حقل العنوان مستعملاً ككود لعملية ثانوية ، مما يؤدي إلى زيادة عدد التعليقات بدون تعديل لحجم الحقل COP . أما الكود الثانوي فيُميز التعليم الخاصة التي تنتمي إلى الفئة المحددة بواسطة الكود الرئيسي .

ملاحظة 5

الحجم (هنا يقاس بعدد المواقع العشرية) للحقل COP يُحدّد العدد الأقصى للدارات - أي للتعليقات الآلية - التي تراجع عنواناً وحيداً يمكن أن تحتويه وحدة الحساب .

5.1 . الحاسب ، العرض الكلاسيكي

بعد هذا المدخل ، نعود إلى عرض أكثر كلاسيكية للحاسب . لقد جرت العادة أن تُميز بين الأعضاء التالية :

الوحدة المركزية وتحتوي :

- الوحدة الجبرية والمنطقية (دارات عمليات ومرافق للحساب) ،

وحدة التحكم وتتألف من :

- مرافق التحكم ،

- عداد البرنامج ،

- الساعة .

الذاكرة المركزية وتتألف من خلايا (كلمات وبيانات) معنونة ،

- أدوات محيطية تسمح بالإدخال والإخراج في الذاكرة المركزية للمعلومات (برامج ومعطيات) المخزنة على نواقل خارجية

فلنذكر البعض منها :

- قارئ البطاقات ، والمثَبّات ، والطابعات ،

- بِسْطَة الأشرطة ، الأسطوانات والطبُول المغناطيسية ،

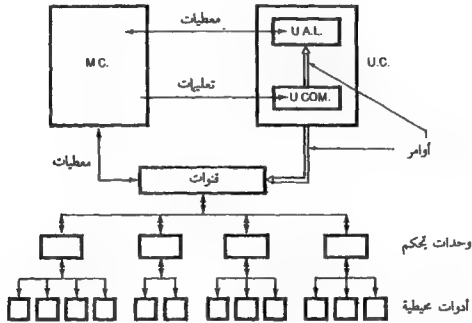
- لوحات ملامس ،- شاشات للعرض ،

- أدوات محيطية خاصة كراسم المنحنيات العاملة حسب النظام «off-line»

(الاشتغال المعزول) .

- القنوات أو وحدات التبادل . وهي عبارة عن الأعضاء التي ، تحت قيادة الوحدة المركزية ، تؤمن بشكل لا تزامني إنتقال المعطيات من الذاكرة المركزية إلى الأدوات المحيطة . هذه الأولوية تسمح بتحرير موارد الوحدة المركزية خلال الوقت ، نسبياً « الطويل » ، للإدخال والإخراج (I/O)⁽¹⁾ . التزامن بين الوحدة المركزية والقنوات (Channels) يتأمن بواسطة نظام الانقطاع الذي سنتكلم عنه لاحقاً .

- وحدة المراقبة والتحكم (Control unit) وهي عبارة عن أجهزة وأدوات ، متكيفة مع كل نوع من المحيطات ، وتحقق عدداً من المهام الضرورية للإدخال والإخراج .



مخطط 6.1

(1) أعضاء الإدخال - الإخراج هي أجهزة إلكترونيكية تمثل إذن نوعاً من القيود . إن قراءة بطاقة معينة قد تطول نحو 100 ميلي ثانية في حين أن وقت تنفيذ تعليمة لا يتعدى أكثر من الميكروثانية (10⁻⁶ ثانية)

2 تكويد المعلومات

الإستعمال الكثير للنظام العشري جعلنا معتادين عليه ، وهذا الإعتياد جعل البعض يخشى من إستعمال نظام آخر للترقيم . ولكن تكنولوجيا الحاسبات تفرض علينا دراسة أنظمة تكويد مختلفة . يجب أن نشير إلى أن التمثيل الثنائي للمعلومات في المكنة لا يحمل أي تعديل لصيغة العمل المشروحة في الفصل الأول ، وهذا من الأسباب التي جعلتنا لا نبدأ الكتاب بهذا الفصل ، راجين أن يكون عرضنا أكثر وضوحاً .

يتألف نظام التكويد من مجموعة قواعد التحويل التي تسمح بالعبور من تمثيل للمعلومات (نص فرنسي مثلاً) إلى ترميز آخر (نص بكود مورس . .) والعكس بالعكس .

الترميز الثنائي هو مفروض لأنه يسمح بتمثيل بسيط لمضمون الذاكرة والمراسف في الحاسب⁽¹⁾ . ويدو أنه لترميز عدد « من حالات صمام كهربائي ، مولع أو مطفا ، فإن التمثيل الثنائي هو الأبسط باعتياد الاتفاق التالي :

1- حالة « الضوء »

0- حالة الإنطفاء

إذاً يرمز إلى الحالة بواسطة :



(1) دون الدخول في التفاصيل التكنولوجية ، تمثل المعلومات داخل الآلة بواسطة عناصر تمتلك حالتين فيزيائيتين مختلفتين .

قد نلاحظ أن مجموعة من صمامين يمكن أن تكون موجودة في عدد $4=2^2$ من الحالات المختلفة التي نرمز إليها على الشكل التالي :

0 0 حالة «0»

0 1 حالة «1»

1 0 حالة «2»

1 1 حالة «3»

ولكن بإمكاننا تكويد :

الحالة «0» : الصمامان هما في حالة الإنطفاء

الحالة «1» : الصمام اليسار هو مفتاحاً ، والصمام الأيمن مؤلّع ، الخ

وبشكل عام ، فإن مجموعة من n من الصمامات يمكن أن تكون موجودة في 2^n حالة مختلفة . يجب تقريب ذلك من الفعل الذي يسمح بواسطة n رقم ثنائي بأن نعد من 0 إلى $2^n - 1$.

1.2 . أنظمة الترقيم :

لو افترضنا أن a_i تمثل مجموعة الرموز المستعملة لتحديد عدد بالقاعدة B ، فإن العدد الحقيقي R يكتب على الشكل التالي :

$$\underbrace{a_n a_{n-1} \dots a_1 a_0}_{\text{القسم الصحيح}} , \underbrace{a_{-1} a_{-2} \dots}_{\text{القسم العشري}}$$

وقيمته هي :

$$R = \underbrace{a_n B^n + a_{n-1} B^{n-1} + \dots + a_0 B^0}_{\text{القسم الصحيح}} + \underbrace{a_{-1} B^{-1} + a_{-2} B^{-2} + \dots}_{\text{القسم العشري}}$$

وفي النظام العشري فإن المجموعة a_i تتألف من الرموز :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

وفي الثنائي : 0 و 1 .

وفي النظام الثماني : 0, 1, 2, 3, 4, 5, 6, 7 .

وفي النظام السادس عشري (16) : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

A, B, C, D, E, F

إنَّ أسَّات القاعدة B^0 ، B^1 ، B^2 ، ... ، B^{-1} ، ... تدعى أوزان الأرقام .
الجدول 1.2 يعطي قيم بعض الأوزان بالنظام العشري :

القاعدة	B^3	B^2	B^1	B^0	B^{-1}	B^{-2}
10	1000	100	10	1	0,1	0,01
2	8	4	2	1	0,5	0,25
8	512	64	8	1	0,125	0,015625
16	4096	256	16	1	0,0625	0,00390625

جدول 1.2

هكذا فالعدد 13 في القاعدة 10 يعادل $(3.10^0 + 1.10^1)$ ويكتب على الشكل التالي : $(1.2^0 + 0.2^1 + 1.2^2 + 1.2^3)$ في النظام الثنائي .

15 في النظام الثنائي : $1.8^1 + 5.8^0$
D في النظام السادس عشري : $(13.16^0 \text{ أي } D.16^0)$
والعدد 0,75 في النظام العشري : $(5.10^{-2} + 7.10^{-1})$
يكتب : 0,11 في النظام الثنائي : $(1.2^{-2} + 1.2^{-1})$
0,6 في النظام الثنائي : (6.8^{-1}) .
و 0,C في النظام السادس عشري : $C.16^{-1}$ أي 12.16^{-1} .

وفي المكنة ، تُعْثَل الأعداد بشكل مكوَّن ثنائيًا . ويمكن أن يحتاج عدد عشري كسري إلى سلسلة طويلة ، أو لا نهائية ، من 0 و 1 . وبما أن الذاكرة والمراصف لها أبعاد محدَّدة عند تصميم المكنة ، لذا ، فقد يحدث تحويل عشري / ثنائي عند الحساب ، أو قد يحدث بتر لقسم من المعلومات مما يؤدي إلى فقدان الدقة في الحساب . وهذه من المشاكل التي يجب الانتباه إليها ولذا من الواجب القيام بعدد كبير من الحسابات التكرارية .

من المهم أن نلاحظ ، أنه عند إزاحة الفاصلة ١٠ موقع لجهة اليسار أو لجهة اليمين فإن هذا يؤدي إلى ضرب العدد أو قسمته على 10 . مثلاً : 13,75 يمثل بواسطة العدد 1101,11 في النظام الثنائي ، ولكن 11011,1 يعادل 27,5 و 110,111 يعادل 6,875 .

ثنائي	سادس عشري	ثنائي	عشري
0	0	0000	0
1	1	0001	1
2	2	0010	2
3	3	0011	3
4	4	0100	4
5	5	0101	5
6	6	0110	6
7	7	0111	7
8	8	1000	8
9	9	1001	9
10	A	1010	10
11	B	1011	11
12	C	1100	12
13	D	1101	13
14	E	1110	14
15	F	1111	15

2.2 جدول

2.2 . تغيير القاعدة

سنترك للقارئ أن يعود للمراجع إذا رغب بذلك . وسنذكر ، بواسطة بعض الأمثلة ، إن التحويلات الثنائية / الثمانية والثنائية / السادس عشرية هي مترابطة لأن القواعد 8 و 16 هي عبارة عن أسس صحيحة للقاعدة 2 .

ينقلب العدد الثنائي إلى سادس عشري بدءاً من كل جهة من موقع الفاصلة . ويتقطع العدد إلى أقسام مؤلفة من أربعة أرقام ثنائية أو بتات⁽¹⁾ ويتأويل كل قسم :

$$\begin{array}{ccccccc} & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & , & 1 & 1 & 0 & 0 & 1 \\ \hline & 2 & & 6 & & B & , & C & & 8 \end{array}$$

ثنائي سادس عشري

الرقم الأخير «8» نحصل عليه بتوسيع الرقم 1 بوضع أصفار لجهة اليمين .

التحويل الثنائي / الثنائي يتم بتقطع العدد الثنائي إلى أقسام مؤلفة من ثلاثة أرقام . نحصل عندها على 62, 1153 في النظام الثنائي .

التحويل العكسي هو بدسجي .

3.2 . الفائلة من النظامين السادس عشري والثاني

سنرى أن كل كلمة آلية هي مكونة من عدد متحول ، يتعلّق بالحاسب ، من العناصر التي تدعى بتات⁽¹⁾ (bit) . كل عنصر يمكن أن يكون موجوداً ، كما هي الحالة

(1) من BIT وهو اختصار للمصطلح الأمريكي Binary digit ، أي رقم ثنائي .

بالنسبة للصمام ، في واحدة من حالتين فيزيائيتين ، لذا يصبح من الطبيعي ترميز حالة البتة بواسطة 0 أو 1 ومضمون الكلمة - الآلية ؛ ليس كما في الفصل الأول بواسطة رقم عشري ، بل بواسطة سلسلة من الأرقام 0 أو 1 ، ويمكن تفسير مجموعة البتات كعدد تمثل في النظام الثنائي .

الأحجام ، المحلدة بعدد البتات ، للكلمات - الآلية التي نلتقيها عادة في الحاسبات هي بطول 8 (الميكروبروسسور) ، 16 ، 24 ، 32 (IBM 360/370) ، 36 ، 48 و 60 بتة . عند تمثيل مضمون كلمة - ذاكرة على ورقة فهذا يتطلب من 16 إلى 60 رمزاً . التمثيل السادس عشري والثاني يظهران إذن مفيدتين مهمين كثيراً لأنها يُقسَّمان على 4 أو على 3 عدد الرموز المطلوب كتابتها وذلك مع المحافظة على إمكانية تحويلها فوراً إلى النظام الثنائي . ولكن النسخ اليدوي لعدد محدد بالنظام السادس عشري هو منيع لعدد أقل من الأخطاء منه في حال كتابته في النظام الثنائي . لذلك فللقارئ فائدة من الإعتياد على هذا النوع من التمثيل المعتمد لتمثيل المعلومات في الذاكرة .

4.2 . الحساب في النظامين الثنائي والسادس عشري
لن نقوم سوى بإعطاء بعض الأمثلة التي يجب أن تسمح للقارئ بإجراء بعض العمليات البسيطة بالجمع والطرح .

في النظام الثنائي :

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ +0 \quad +1 \quad +0 \quad +1 \\ \hline 0 \quad 1 \quad 1 \quad 10 \end{array}$$

$$\begin{array}{r} 1101 \\ +111 \\ \hline 10100 \end{array}$$

مثلاً :

$$\begin{array}{r} D \\ +7 \\ \hline 14 \end{array}$$

في النظام السادس عشري :

في النظام السادس عشري من العملي تحويل كل رقم إلى النظام العشري ، وإجراء العملية في هذا النظام ومن ثم تحويل النتيجة . مثلاً :
 $D_{16}=13_{10}$ ، $7_{16}=7_{10}$ ، $13+7=20_{10}=16+4$: نضع 4 ونحفظ باليد 1 ، إلخ :

$$\begin{array}{r} 3F2 \\ +1A4 \\ \hline 596 \end{array} \quad \begin{array}{r} 3F2 \\ -1A4 \\ \hline 24E \end{array}$$

بنفس الطريقة نقوم بإجراء الطرح 4 - 216 تصبح 4-2+16 أي E وباليد 1 . . .
حسب نفس الصيغة سنستطيع إجراء الحساب في النظام الثنائي . وبإستطاعة
القارئ أن يتحرر بوجود الأمثلة المعطاة في نهاية الفصل .

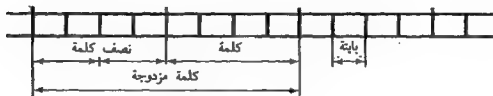
5.2 . التمثيل الداخلي للمعطيات

1.5.2 . الذاكرة

حتى هذا الوقت إعتبرنا إن الذاكرة هي مؤلفة من خلايا مرقمة بدءاً من 0 ،
الخلية هي الكلمة - الآلية والعناوين هي عناوين الكلمات .

سنقوم بتحديد الأشياء . المكتات IBM 360/370 تتمتع بكلمة - آلية من 32 بتة
مرقمة من اليسار إلى اليمين من 0 إلى 31 . تُقسّم الكلمة إلى أربع بايتات (تشكيلة من
8 بتات) . والبايتة هي قابلة للعنونة . ستكلم عن الذاكرة المعنونة بالسيات (وسمى
إن ألسمة قابلة للتمثيل بواسطة 8 بتات) مقابلة مع بعض المكتات حيث الذاكرة معنونة
بالكلمات . عنوان الكلمة هو إذاً عنوان البايته الأولى من الكلمة . في النهاية نوجز ما
يلي :

- جهات النصف كلمات هي بعناوين مزدوجة ؛
 - جهات الكلمات هي بعناوين قابلة للقسمه على أربعة ؛
 - جهات الكلمات المزدوجة تتمتع بعناوين قابلة للقسمه على 8 ؛
- ومع إن الذاكرة هي قابلة للعنونة في مستوى البايته ، يجب السهر على المحافظة
على هذا التقسيم للمعطيات المثلة بواسطة نصف كلمة ، كلمة ، أو كلمة مزدوجة .



شكل 3.2

2.5.2 . تمثيل المعطيات الالرقمية

بإمكاننا تكويد نوعين من المعلومات في الذاكرة : المعطيات الرقمية والتي هي
عبارة عن تشكيلات ثنائية مرتبطة بمعنى رقمي ، والمعطيات من نوع سيات والمعالجة
كوححدات غير رقمية .

لقد كان من الملائم عند تصوّر مكنات IBM 360/370 ، تكويد السيات بواسطة 8 بتات . هذا النظام يسمح بتكويد 2^8 ، أي ما مجموعه 256 كوداً مختلفاً . هذا التصوّر هو واسع الإنتشار ، ولكن هناك مكنات أخرى تستعمل تكويد السيات بواسطة 6 بتات -تُحلّد مجموعة السيات المتوفّرة بالعدد 64 سمة .

قد يبدو لنا مفاجئاً إعتداد كود لتمثيل السيات بواسطة 8 بتات . فلنلاحظ ببساطة إن هذا النظام يسمح لنا بالحصول على ألفباء واسعة تحتوي على السيات الكبيرة ، والصغيرة ، والسيات العشر العشرية وبعض السيات الخاصة ، كإشارات العمليات ، وعلامات الوقف ، والفسحة ، الخ .

الكود الداخلي لتمثيل السيات ، والمستعمل على المكنات IBM 360/370 هو EBCDIC (Extended Binary Coded Decimal Interchange Code) . يُرمز إلى الحرف A بواسطة الكود 11000001 ، أي C1 بالترميز السادس عشري . ويكوّد الحرف «B» بواسطة C2 وهكذا دواليك . لائحة الأكواد موجودة في الملحق . مثلاً : لنفترض إن مضمون حيزّ الذاكرة هو التالي :

0	0	C	1	E	2	F	2	C	5	D	4	E	2	D	3	C	5	E	4	D	9	4	0	F	3	F	7	F	0	0	0
100								104								108								112							

تأويل هذه السلسلة من 14 بايتة ، والتي تبدأ بالعنوان 100 ، هو حسب الكود «ASSEMBLER 370» .

نشر إلى وجود علاقة تراتبية بين القيم الثنائية المستعملة للتكويد :

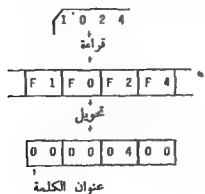
$$40 < C1 < C2 < \dots < F0 < F1 < \dots < F9$$

وهذا ممكن أن يترجم بواسطة :

كود الأرقام > ... > كود B > كود A > كود القسمة .

هذه الخصوصية هي مستعملة للترتيب الأبجدي .

يجب أن نُميّز بين التمثيل الأبجدي والتمثيل الرقمي . المثل التالي يوضح لنا التحويل المعتمد لمعطى مقروء من البطاقة ومحوّل إلى ثنائي .



ذاكرة (حيز إدخال)

تمثيل بطريقة السيات

ذاكرة (متحوّلة مؤشّرة في)

FORTRAN في لائحة الأمر (READ) .

تمثيل بطريقة الفاصلة الثابتة

التمثيل السبائي يُقال عنه أيضاً « القابل للتتبع » لأنه ضمن هذا الشكل يجب أن تكون المعلومات موجودة قبل أن تستلمها الطابعة لطبعها .

3.5.2 . تمثيل المعطيات الرقمية

المعتادون على لغة فورتران يعلمون أن المتحولة أو الثابتة يجب أن تُمثل دائماً في المكتبة بواسطة كلمة (أو كلمة مزدوجة عندما يكون الحيزُ مصرحاً عنه بدقة مزدوجة) . ويعلمون أيضاً إن هذه اللغة تستعمل نوعين أساسيين من التمثيل الداخلي للمعطيات الرقمية : النوع الصحيح (integer) والنوع العائم (real) .

أما المعتادون على لغة كوبول فلا يجهلون أن الحسابات الجارية بهذه اللغة تتم بواسطة تمثيل مجهول من لغة فورتران : التمثيل العشري المتراص . سنجد هذه الطرق الأربع في تكويد الأعداد في مستوى المكتبة : الطريقة « الفاصلة الثابتة » (fixed point) « صحيح ببلغة فورتران » ، والعائم البسيط والعائم الموسع والصيغة العشرية المتراصة . نشير إلى أن مع كل نوع من هذه التمثيلات تتلامم مجموعة من المؤثرات (دارات الكترونية ، + ، - ، ...) ، صالحة للعمل بهذه التشكيلات الثنائية . وفي النتيجة فإن المكتبات تحتوي على أربع مجموعات من التعليمات الجبرية .

أ - التمثيل بفاصلة ثابتة

بهذه التسمية يجب أن نفهم « فاصلة ثابتة إتفاقياً » . هكذا ، فالفاصلة ، عنصر أساسي من قيمة العدد ، لا تظهر أبداً في التمثيل الداخلي للعدد في الذاكرة . ولقد لاحظنا (في الفقرة 1.2) أن التشكيلات الثنائية المعتمدة لـ n ، $2n$ ، $n/2$ لا تختلف إلا بواسطة موقع الفاصلة ، لذا ، فإن 1001 يمكن أن تُمثل القيمة 9 إذا اعتبرنا أن الفاصلة موجودة لجهة اليمين ، أو 0.5625 إذا اعتبرنا أن الفاصلة موجودة في أقصى اليسار النظام IBM 360/370 يفترض الفاصلة موضوعة لجهة اليمين . وللتأكد من ذلك يكفي ملاحظة التعليقات التي تسمح بجمع المعطيات بطول مختلف (كلمة أو نصف كلمة) . إن عملية التسطير للمعلومات تتم لجهة اليمين . هذا التمثيل هو إذاً التمثيل الصحيح . وهناك بعض المصممين الآخرين الذين اعتمدوا الإتفاق الماكس ، أي الفاصلة لجهة اليسار .

تُكوّد الأعداد حسب النظام الثنائي في كلمة - آلية . البتة ذات الوزن الأكبر (البتة الموجودة لجهة اليسار) ترمز إلى الإشارة الحسابية . إذا كانت تساوي 0 ، يكون العدد إيجابياً ، أما إذا كانت تعادل 1 فمعنى ذلك أن العدد هو سلبى .

بواسطة n بتة باستطاعتنا تعداد من 0 حتى $2^n - 1$. وإذا حجزنا بتة للإشارة فسيكون بإمكاننا تمثيل الأعداد الصحيحة I بحيث إن :

$$-2^{n-1} \leq I \leq 2^{n-1} - 1$$

إذا كانت $n = 16$: $32767 \leq I \leq 32768 -$

تمثيل الأعداد الإيجابية لا يفترض أية مشكلة ، والتأويل العشري نحصل عليه بضرب كل بته بالوزن المعتمد للموقع . وفي المقابل يجب أن نعتد إتفاقاً جديداً للأعداد السلبية .

تمثيل الإشارة والقيمة المطلقة
الفكرة التي نخطر لنا تقوم على اعتبار البتة ذات الوزن الأقوى ترمز إلى الإشارة والباقي يرمز إلى القيمة المطلقة للعدد . حسب هذا الإتفاق ، الممثل بأربع بتات :

$$\begin{array}{r} 0101 \\ 1101 \\ \hline 10010 \end{array} \quad \begin{array}{l} + \text{ يكتب :} \\ - \text{ يكتب :} \\ \text{نتيجة الجمع :} \end{array}$$

هذه النتيجة هي ليست حقيقية .

هذا التمثيل يُحتم علينا إذاً ، للحصول على النتيجة الصحيحة ، أن نفحص الإشارات المرتبطة بالتأثيرات قبل إجراء العمليات . لا يجب معالجة الأعداد السلبية والإيجابية بنفس الطريقة . يمكن للقارىء أن يقتنع بأن إعتاد هذه الصيغة يحتم علينا إعتاد منطق الكتروني أكثر تعقيداً . وقد جرى التخلي عنه اليوم .

التمثيل المدعو مُكمّل 1 (1 Complement)

عكس العدد (أو ضده) . نحصل عليه بأخذ عكس كل بته . بما فيها بته الإشارة . هكذا :

$$\begin{array}{r} 0101 \\ 1010 \\ \hline 1111 \end{array} \quad \begin{array}{l} + \text{ تكتب :} \\ - \text{ يكتب :} \end{array}$$

وبنتيجة الجمع نحصل على

أي ، أن مُكمّل 1 هو 0000

من الممكن إعتبار إن هذا النوع من التمثيل يؤدي إلى إدخال 0 إيجابي وصفر سلمي . المسائل المطروحة في نهاية الفصل تشرح سيئات هذه الاتفاقات وفوائد الاتفاقات اللاحقة

التمثيل المدعو « مُكمّل إلى 2 » (Two complement)

هو التمثيل المعتمد على المكنات IBM 360/370 . يُمثل كل عدد سلمي بواسطة المُكمّل إلى 2^n لعكس العدد . ولو افترضنا إن X هو العدد ، وأن \bar{X} هو مُكمّل العدد X إلى 2^n ، نحصل إذاً على العلاقة التالية $X + \bar{X} = 2^n$. الإتفاق حول الإشارة هو كالمسابق . ونشير إلى أن المعطيات الرقمية هي مكوّدة بأطوال ثابتة ، هي الكليات -

الآلية . وللمكنات IBM 360/370 ، n تعادل 32 . ولتسهيل العمل ، فإننا منعالج مسائل تعمل بأربع أو ثمان بتات .

وبالتكويذ بواسطة أربع بتات ، حيث البتة اليسرى هي بتة الإشارة ، فإن كود العدد -5 هو المُعادل الثنائي لـ $2^4 - 5 = 11$ إذن :

$$\begin{array}{r} +5 \\ -5 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0101 \\ 1011 \\ \hline 10000 \end{array}$$

وبإهمال الحاصل بعد موقع الإشارة نحصل على صفر .

الطريقة للحصول على المُكْمَل إلى 2 لعدد ما تكمن بتكملة العدد إلى 1 وبعد ذلك إضافة 1 إليه . تتم العمليات على جميع البتات بما فيها بتة الإشارة .
مثلاً :

$$\begin{array}{r} +5 \\ \text{مُكْمَل إلى 1} \\ +1 \\ \hline 0101 \\ 1010 \\ +1 \\ \hline 1011 \end{array}$$

فلنلاحظ إنه إذا كنا نعمل على عدد ثنائي مُمَثَّل بالتريقيم السادس عشري ، فإن المُكْمَل إلى 2^n يصبح مُكْمَلًا إلى 16^p . سنحصل على التمثيل السادس عشري للعدد للعكوس بتكملة كل رقم إلى F وإضافة 1 .
مثلاً : جل ثمان بتات :

$$\begin{array}{r} 0100 \ 1101 \rightarrow 4D \\ + \\ B2 \\ +1 \\ \hline 1011 \ 0011 \leftarrow B3 \end{array}$$

انتقال العدد ، المُمَثَّل بواسطة 16 بتة ، في مرصف بطول 32 بتة سيتم بواسطة انتقال بسيط إلى اليسار للبتة ذات الوزن الأكبر :

$$\begin{array}{r} 0A1C \rightarrow 0000 \ 0A1C \\ B0D3 \rightarrow FFFF \ B0D3 \end{array}$$

حالة الفيض عن السعة (Over flow) ، يمكن أن تحدث عند إجراء عملية معينة وذلك عندما يكون كلا المتأثرين بنفس الإشارة والنتيجة تصبح بإشارة معاكسة .
لنحيط بعض الأمثلة على معطيات ممثلة بواسطة أربع بتات . مجموعة الأعداد القابلة للتمثيل هي :

1111	-1	0000	0
1110	-2	0001	+1
1101	-3	0010	+2
1100	-4	0011	+3
1011	-5	0100	+4
1010	-6	0101	+5
1001	-7	0110	+6
1000	-8	0111	+7

بالإمكان إهمال المرحّل اليسار بدءاً من موقع الإشارة ، إذا كان كلا المتأثرين بنفس الإشارة ، والنتيجة بنفس الإشارة .

وجود المرحّل ، ويُسمى (Carry) في المصطلحات الأنكلوسكسونية ، ليس هو إشارة خطأ في الحساب .

سنلاحظ في النهاية إن العدد الأصغر القابل للتمثيل هو 2^{-1} والأكبر هو $1 - 2^{-1}$ وإن الطرح يمكن أن يتم بواسطة الجمع إلى مكمّل 2 .

نتيجة		مرحّل مفقود	عشري
خطأ ODC (1)	$\begin{array}{r} 0\ 111 \\ 0\ 111 \\ \hline 1\ 110 \end{array}$		$\begin{array}{r} +7 \\ +7 \\ \hline 14 \end{array}$
خطأ ODC (1)	$\begin{array}{r} 0\ 100 \\ 0\ 101 \\ \hline 1\ 001 \end{array}$		$\begin{array}{r} +4 \\ +5 \\ \hline 9 \end{array}$
صحيح	$\begin{array}{r} 0\ 100 \\ 1\ 011 \\ \hline 1\ 111 \end{array}$		$\begin{array}{r} +4 \\ -5 \\ \hline -1 \end{array}$
خطأ ODC (1)	$\begin{array}{r} 1\ 100 \\ 1\ 011 \\ 0\ 111 \end{array}$	1	$\begin{array}{r} -4 \\ -5 \\ \hline -9 \end{array}$
صحيح	$\begin{array}{r} 1\ 101 \\ 0\ 011 \\ 0\ 000 \end{array}$	1	$\begin{array}{r} -3 \\ +3 \\ \hline 0 \end{array}$

(1) ODC : الفيض عن السعة (over flow)

ب - التمثيل بفاصلة متحركة

الحساب العلمي يستعمل عادة أعداداً بأحجام كبيرة جداً أو صغيرة جداً ولكن ممثلة بواسطة عدد مُحدّد من الأرقام . النوع فاصلة ثابتة لا يسمح بالتمثيل البسيط لهذه الأعداد ، ولذلك إعتدنا طريقة أخرى في التكويد المركّب من قسمين :

- المُميّزة (الأسّ المُعيّن) التي تعطي الحجم .
- القسم العشري (mantisse) الذي يحدد الأرقام ذات الأوزان الكبرى .

هكذا فيإمكاننا تحديد العدد على الشكل التالي :

S.M.B^c

حيث S هي الإشارة ، M القسم العشري (mantisse) ، B عدد ثابت (2 ، 10 ، أو 16 حسب المكتة) ، C هي الأسّ المُعيّن .

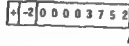
كما في الفاصلة الثابتة ، فإن الفاصلة لا تظهر في التكويد الداخلي ولكنها توضع عادة إلى يمين أو إلى يسار القسم العشري M . هكذا ، فلنظام بقاعدة B=10 ، يُكتب العدد 37,52 على الشكل التالي⁽¹⁾ :

1 - الفاصلة لجهة يمين القسم العشري .

$$(1) \quad 37520.10^{-3}$$



$$(2) \quad 3752.10^{-2}$$



2 - الفاصلة إلى يسار القسم العشري ،

$$(3) \quad 0,000003752.10^7$$



$$(4) \quad 0,3752.10^2$$



نلاحظ ، في الحالة التي تكون فيها الفاصلة موجودة إلى يسار القسم العشري ، إن التمثيل (4) يعطي عدداً أكبر من الأرقام ذات المعنى (Significants digits) من التمثيل

(1) Δ : رمز يشير إلى موقع الفاصلة .

(3) ، في الحالة التي يكون فيها عدد الأرقام المحجوزة للقسم العشري ثابتاً . التمثيل (4) يُدعى موحد التنظيم المعايير (normalized) . وهو يتناسب مع حصر الأرقام ذات المعنى من القسم العشري لجهة اليسار . هذا التمثيل يسمح بأكثر دقة ممكنة . من الممكن أن نغير من تمثيل معين إلى تمثيل معايير آخر بواسطة إزاحة الأرقام وتعديل الأس .

إذا كانت $B = 10$ ، فإن الإزاحة إلى اليسار لموقع رقم يؤدي إلى تنقيص الأس المعين 1 .

أما إذا كانت $B = 16$ ، فإن الإزاحة إلى اليسار لرقم سادس عشري من القسم العشري سيؤدي إلى تنقيص 1 من الأس المعين . وهكذا سيكون العدد عملاً بشكل معايير عندما لا يكون الرقم السادس عشري ذو الوزن الأكبر من القسم العشري صفراً . سنشير إلى أن الإزاحة لموقع سادس عشري يُترجم إزاحة أربعة مواقع ثنائية . على الحاسبات IBM 360/370 :

- الإشارة S من العدد هي مكوّنة على بته واحدة ($0 = +$ ، $1 = -$) ؛
- القاعدة B هي 16 ؛

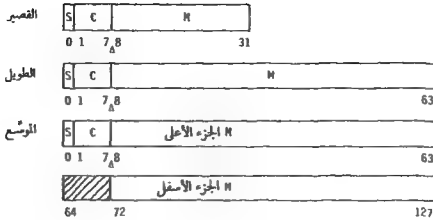
- يفترض أن تكون الفاصلة إلى يسار القسم العشري الذي يُمثّل عدداً أصغر من 1 .
- العدد الثنائي المكوّن في الحيز C بطول 7 بتات والمحفوظ للأس المعين ، لا يُمثّل أبداً قيمة الأس المعين E إلا 16 ولكن :

$$C = 64_{16} + E$$

لذا فهناك مشكلة في توكيد إشارة الأس كي نحصل على قوى سلبية وإيجابية للقاعدة B بدلاً من اعتماد ترميز شبيه بالمكمل إلى 2 ، لقد جرى إختيار اعتياد القوة صفر في التوكيد المناسب للقيمة الوسطية للأعداد القصوى 0 و $2^7 - 1$ أي 64_{16} أو 40_{16} أو 1000000_{10} . هكذا ، عندما تكون $C = 64_{16}$ فإن قيمة العدد هي $S.16^0.M$ ، وعندما تكون $C > 64$ ($C = 64 + E$) فإن قيمة العدد هي $C.S.16^E.M$ متغيرة من 0 إلى 127 وبالتالي $-64 \leq E \leq +63$.

للحصول على E يكفي ، في النظام السادس عشري ، أن نطرح 40_{16} : 40_{16} تناسب $E = 6$ و $3F_{16}$ تناسب $E = -1$.

يوجد على الحاسبات IBM 370 ثلاثة أشكال بفاصلة متحركة . الأعداد بالفاصلة المتحركة الصغيرة تحتل كلمة - آلية ، والأعداد الطويلة تحتل كلمتين - آليتين ، والأعداد الموسّعة تشغل أربع كلمات . الشكل الأخير هو غير موجود على المكنات 360 .



4.2 خطط

الأشكال الثلاثة تسمح بتكويد أعداد بنفس الحجم . وتختلف بواسطة عدد الأرقام ذات المعنى التي تقدمها . العدد P هو :
بالشكل القصير :

$$16^{-63} < P < (1 - 16^{-6}) \cdot 16^{63}$$

7 أرقام عشرية ذات معنى

$$16^{-63} < P < (1 - 16^{-14}) \cdot 16^{63}$$

- بالشكل الطويل

16 رقماً عشرياً ذا معنى .

$$16^{-63} < P < (1 - 16^{-28}) \cdot 16^{63}$$

- بالشكل الموسع

33 رقماً عشرياً ذا معنى .

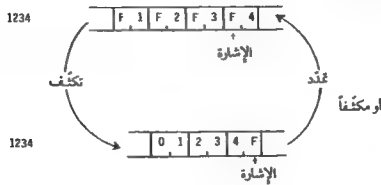
وفي الحالات الثلاث يكون معنا تقريباً : $5,4 \cdot 10^{-79} < P < 7,2 \cdot 10^{75}$

أما الحسابات بواسطة هذه الطرق في التمثيل فقد تؤدي إلى فيض عن السعة (Overflow) عندما نحصل على قيم كبيرة جداً أو صغيرة ونُدعى Overflow أو Underflow للأعداد بالفاصلة المتحركة .

مثلاً : التمثيل بفاصلة متحركة

C 2 1 9 0 0 0 0	-25	$-(1.16^{-1} + 9.16^{-2}) 16^2$
C 1 1 0 0 0 0 0	-1	$-(1.16^{-1}) 16^1$
0 0 0 0 0 0 0 0	0	$0 . 16^{-64}$

ج - التمثيل العشري
يمكن أن يتم تمثيل العدد بواسطة النظام العشري المكوّن ثنائياً (DCB) موسّعاً ،
أي على شكل سمات .



هذا التمثيل ، الواسع الإنتشار في الإدارة ، هو أقل « تراصاً » من سوابقه . لا يوجد أي طول ضمني لها : توضع المعطيات بداخل بايتات . سنرى ان التعليلات PACK و UNPK تسمح بالعبور من شكل إلى آخر .
الفاصلة ، كما رأينا ، ليست مُمثلة . وإن تنظيم موقعها والاصطفاف المحتمل المناسب يقع على عاتق المبرمج . ونشير إلى المواقع المختلفة للإشارة . القيم السادس عشرية E, C, A, يمرري تأويلها وكأنها « + » . أما B و D فيؤوّلان وكأنهما « - » .

تمارين

تمرين 1.2 - غيّر إلى النظام الثنائي والسادس عشري ، الأعداد العشرية التالية :
15 35 256 1024 348,5 .

تمرين 2.2 - غيّر إلى النظام العشري والثنائي الأعداد السادس عشرية التالية :
3A FFF 1A3B ABC

تمرين 3.2 - إحصب المكمل إلى 2 للعدد 1A3B . أطرح 1A3B من العدد 2ABC .
أعط التمثيل الموسّع إلى 32 بتة للعدد 1A3B وكذلك لمكمله إلى 2 .

تمرين 4.2 - أعط القيم الرقمية العشرية التي نقوم بتأويلها :
C1FD0000

- كمعطى ممثّل بفاصلة ثابتة حسب توكويد الاشارة والقيمة المطلقة .
- كعدد ممثّل بالمكمل إلى 2 .
- كعدد بفاصلة متحركة بطول قصير (هل هو معايير في هذه الحالة؟) .
هل بالإمكان اعتبار هذا التشكيل الثنائي كمعطى مكوّد بالشكل العشري ؟

ما هو نقيض (أو ضدّ) هذا العدد في كلّ من التمثيلات المذكورة ؟

تمرين 5.2 - عاير العدد بفاصلة متحركة C5032000 .

3 . العنوان المطلقة ، العنوان النسبية

1.3 . عموميات

في الفصل الأول عرضنا التعليمات - الآلية وكأنها مشكلة من حقليين : الحقل كود العملية (operation code) وحقل العنوان . تحتوي التعليمة إذا على العنوان المطلق للمتأثر ، أي عنوانه الفعلي أو الحقيقي بالنسبة للعنوان 0 من الذاكرة . هكذا فبرنامج جمع مضمون الجلايا 0 و1 وخزن النتيجة في العنوان 2 كان قد كتب على الشكل التالي:-

0		المتأثر الأول
1		المتأثر الثاني
2		النتيجة
3	8 8 0	$(0) + AQ$
4	9 0 1	$AQ+1 + AQ$
	8 0 2	$AQ + (2)$

فلنفترض بأننا زرعنا هذا البرنامج (مجموعة مناطق العمل والتعليمات) ليس على العنوان 0 ولكن على العنوان 100 . سنكتب عند ذلك :

100		المتأثر الأول
101		المتأثر الثاني
102		النتيجة
103	8 8 1 0 0	$(100) + AQ$
104	9 0 1 0 1	$AQ+(101) + AQ$
105	8 0 1 0 2	$AQ + (102)$

نلاحظ أن كود العمليات لا يتغير ولكن العناوين قد جرى نقلها 100 موقع لأن التعليقات تعود إلى العناوين المطلقة . أو بشكل آخر ، فإن كتابة البرنامج تتعلق بالعنوان الفعلي لمكان البرنامج . هذا الإلزام ، الذي ستعرض سببته ، قد أجبر مُصممي المكونات على تعريف أولية العنونة النسبية : حقل العنوان من التعليمة لا يعود إلى العنوان المطلق للمتأثر ولكن إلى عنوان نسبي حسب عنوان أساسي (مطلق) . وبالإجمال فإن حقل العنوان يعطي « المسافة » إلى موقع المتأثر بالنسبة إلى عنوان يُعتبر وكأنه أساس أو قاعدة (Base adresse) ويعرف في لحظة زرع البرنامج في الذاكرة . العنوان الفعلي (المطلق) للمتأثر سيحسب ، في لحظة تنفيذ التعليمة ، بواسطة جمع العنوان المرجعي (الأساسي) إلى قيمة الإزاحة المحددة في التعليمة .
سنعتمد في ما يلي إلى شرح أواليات عدّة للعنونة تتواجد في نفس الوقت على الآلات الحالية .

2.3 . العنونة القاعدية

هي عنونة نسبية حيث المبدأ هو كما ورد أعلاه . يحتوي الحاسب على عدد من المرافص التي يمكن أن تستعمل كمراصف أساسية (قاعدية) ، ويجب على المبرمج :
- أن يختار المرفص الأساسي بواسطة أمر خاص .
- أن يُخزّن قيمة معينة في هذا المرفص ، قيمة ستكون عبارة عن العنوان الأساسي .
- كتابة البرنامج (معطيات وتعليقات) نسبة إلى عنوان معين يعادل عادة الصفر .
وفي لحظة التنفيذ يُسحّن البرنامج في الذاكرة ، وتُخزّن قيمة العنوان القاعدي في المرفص القاعدي . عند تنفيذ كل تعليمة فإن العنوان الموجود في التعليمة (الإزاحة (déplacement) يُضاف أوتوماتيكياً إلى مضمون المرفص القاعدي للحصول على العنوان الفعلي للمتأثر .

ذاكرة

1 5 0

المرفص القاعدي

150	
151	
152	
153	8 8 0
154	9 0 1
155	8 0 2
156	

المتأثر الأول

المتأثر الثاني

النتيجة

يكتب البرنامج دون الإهتمام بالعنوان الفعلي لمكان تخزين البرنامج . وتُحسب جميع العناوين نسبة إلى العنوان صفر (بداية البرنامج) .

ولنفترض إن بداية البرنامج (العنوان النسبي صفر) موجودة على العنوان الفعلي 150 ، وهي قيمة سيتم تخزينها في مرصف القاعدة⁽¹⁾ . إذا فالعنوان النسبي n للبرنامج يناسب العنوان الفعلي $n + 150$. . . والبرنامج سيقوم بتنفيذ العملية :
 $(152) \rightarrow (151) + (150)$

لدينا إذن العلاقة التالية :

العنوان الفعلي = العنوان القاعدي + العنوان الموجود في التعليمات

نشير إلى أن عملية الجمع تتم ديناميكياً ، في لحظة تنفيذ كل تعليمات . يبدو من البديهي أن المبرمج لا يجب أن يُعَدِّل مضمون المرصف القاعدي . العنوان النسبي الموجود في التعليمات يُدعى إزاحة (déplacement) .

المكنات IBM 360/370 تتمتع بـ 16 مرصفاً عاماً يمكن أن تُستعمل كمراسف قاعدية . يُحَدِّد المرصف بالكامل بواسطة رقم المرصف المستعمل كمرصف قاعدي والعنوان النسبي . هكذا ، فإن حقل العنوان من تعليمات هذه المكنات سيحتوي على حيز من أربع بتات حيث يتم تخزين رقم مرصف القاعدة .
 الحسّنات :

- يكتب المبرمج برنامجاً بشكل مستقل عن الموقع الذي سيشغله في داخل الذاكرة .
- البرنامج ، أو مجموعة الحيزات والتعليمات ، هو قابل للتحويل والنقل . من الممكن نقله من حيز من الذاكرة إلى حيز آخر دون تعديل في العناوين المنقولة (المحوّل) . يكفي تعديل مضمون المرصف القاعدي .
- العنونة الأساسية وبشكل عام العنونة النسبية تسمح بعنونة مناطق كبيرة من الذاكرة بدلاً من أن تحتوي التعليمات ، على حقل عنوان طويل جداً . نشير حول هذا الموضوع ، أنه لعنونة 2^n خلية من الذاكرة يلزمنا عدد n من البتات .
 السيئات :
- كل تعديل في مرصف القاعدة خلال تنفيذ التعليمات يؤدي إلى نتائج غير متوقعة .

3.3 . العنونة المؤشرة (Indexed address)

يتعلّق ذلك بعملية حسابة العنوان بشكل شبيه بالعنونة القاعدية ولكن بهدف مختلف . يوجد مرصف يدعى مرصف التأشير أو مرصف الدليل (index register) ،

(1) العنوان القاعدي ليس بالضرورة عنوان زرع البرنامج .

تُخزَّن فيه قيمة معينة بواسطة المبرمج :
هكذا :

العنوان الفعلي = العنوان القاعدي + الإزاحة + مضمون المرفص المؤشر

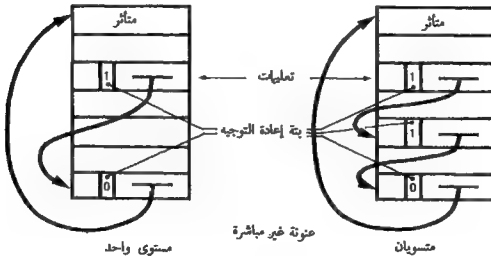
وعلى عكس مرفص القاعدة ، فإن مرفص التأشير يُمكن أن يُعدَّل مضمونه بواسطة المبرمج . هذه الأوالية تسمح ، بواسطة عمليات الزيادة على مضمونه هذه، بأن نقوم بعمليات تكرارية ، وتشكيل حلقات (loop) من التعليمات ، وبالتالي بلوغ خلايا متتالية من الذاكرة . هذه هي التقنية المستعملة لبلوغ الجداول . التعليمات التي تعود إلى عناوين والتي يُمكن أن تحتل عملية تأشير تتمتع بحقل إضافي خاص بالمرفص المؤشر حيث يستطيع المبرمج وضع رقم المرفص الذي يرغب باستعماله كدليل أو كمؤشر (index) .

4.3 . العنوان المباشرة

نتكلم عن العنوان المباشرة عندما نجد في التعليمة العنوان الفعلي للمتأثر . إنها إذن أوالية العنوان البسيطة والمطلقة .

5.3 . العنوان غير المباشرة

هذه التقنية في العنوان موجودة على أكثر المكونات-حقل العنوان من التعليمة لا يحتوي على عنوان المتأثر ولكن على كلمة تحتوي عنوان المتأثر . بعض المكونات تتمتع ، عتادياً ، بأداة خاصة لتغيير الاتجاه . في هذه الحالة ، يوجد بة خاصة في التعليمة تشير إلى وجود أو عدم وجود إعادة تغيير في الاتجاه . إعادة التوجيه يمكن أن تتم في مستويات عديدة كما يبرهن لنا المثل التالي :



6.3 . العنوان التلقائية

هَذَا المصطلح الشائع هو سيء لأن هذه الطريقة لا تخص عنواناً معيناً إنما تخص قيمة محدّدة . المعلومة الموجودة في حقل التعليمة المُستعمل لكتابة العنوان ، لا تُثْمَل عنوان المتأثر ، إنما المتأثر نفسه (قيمة تستعملها التعليمة) .

تصغير المرصف يمكن أن يتم بطريقتين :

- بواسطة العنوان المباشرة يتم تصغير كلمة من الذاكرة بعنوان A ، وسنستعمل تعليمة لشحن المرصف بعنوان مباشر مع مضمون $R:A \rightarrow (A)$ ؛

- بواسطة العنوان التلقائية ، سيجري نقل القيمة صفر الموجودة في التعليمة على موقع العنوان إلى المرصف مع احتمال إزاحة البتة ذات الوزن الأكبر إلى اليسار إذا كان حجم حقل العنوان أصغر من حجم المرصف . العملية تتم بدون مساعدة أية خلية إضافية من الذاكرة . الحاسبات IBM 360/370 تتمتع بمجموعة من التعليمات ، تلك ذات الصيغة SI ، وتعمل بعنوان تلقائية .

4 هيكلة الحاسبات 370 / IBM 360

لن نقوم هنا سوى بإيجاز المميزات الضرورية الواجب معرفتها للبرمجة . بعض النقاط يمكن أن تعتبر حاجزاً أمام القارئ المبتدئ ، ومستوضح له لاحقاً إلا أننا وجدنا من المفيد تحديدها منذ الآن .

1.4 . الذاكرة

الذاكرة هي معنونة بالبايتات (فقرة 1.5.2) . وسعتها القصوى هي 16777216 بايتة (2²⁴) . تُرُقَّم البايتات على التوالي بدءاً من الصفر . تجري التعليقات على سلاسل من البايتات ، نصف كلمات (عناوين مزدوجة) من بايتين ، وكلمات (عناوين تقبل القسمة على 4) من أربع بايتات وكلمات مزدوجة (عناوين مضاعفة لـ 8) من ثمان بايتات . تُرُقَّم بتات الكلمات من اليسار إلى اليمين من 0 إلى 31 .

2.4 . المرافص

تستعمل مرافص التحكم بواسطة نظام التشغيل لإدارة الذاكرة . وهي مبلوغة بواسطة تعليقات مميزة وخاصة ، لن نتكلم عنها .

المرافص العامة وعددها 16 ومُرَقَّمة من 0 إلى 15 ، ويمكن أن تُستعمل :
- كمرافص قاعدية (أساسية) (ما عدا المرصف 0) ، وتحتوي على عنوان مطلق من 24 بتة من اليمين .

- مرافص دلالية (مرصف مؤشر) (index register) (ما عدا المرصف رقم 0) .
- مرصف شحن (مركم) أو توسيع لمرصف الشحن يستعمل لإجراء العمليات على التمثيلات الداخلية للأعداد بفاصلة ثابتة أو عمليات منطقية . بعض العمليات تحتاج إلى وجود مرصفين «متلاحقين» (الضرب مثلاً) . تستعمل عندئذٍ مرافص عامة متتالية ، الأول يكون إلزامياً برقم مزدوج . سنُسَمِّي لاحقاً زوجاً من المرافص كهذا ، مرصفاً مزدوجاً . التعليقات التي تستعمل مرصفاً مزدوجاً لا تشير سوى إلى المرصف برقم مزدوج .

- الأتمة . وهي مرتبطة بمختلف أسباب الانقطاعات . وجود البتة «0» في بتة القناع يمنع المعالجة المباشرة للحادثة . الإنقطاعات من نوع overflow (قناع البرنامج) يمكن أن تحمل ، وتوضع الأخرى في الانتظار حتى رفع أوزوال سبب المنع أو الإهمال . فقط بإمكان المبرمج بلوغ قناع البرنامج عندما يعمل الأخير في صيغة المسألة (15=bit) البتة رقم 15 تعادل 1 .

البتات من 0 إلى 6 تتعلق بالإنقطاعات الآتية من القنوات . البتة 7 (E) ، الانقطاعات الخارجية ، البتة 13 (M) ، عمل المكنة السيء والبتات من 36 إلى 39 ، الانقطاعات الناتجة عن تجاوز في السعة ، البتة 36 مرتبطة بالفيض عن السعة (Overflow) أثناء إجراء العمليات الجبرية بفاصلة ثابتة ، والبتة 37 متعلقة بالنظام العشري والبتتان 38 و39 متعلقتان بالحساب بفاصلة متحركة .

- مفتاح الحماية : هذا المؤشر (البتات من 8 إلى 11) ، وبالعلاقة مع المفتاح الموجود في الذاكرة ، يتيح أو يمنع بلوغ البرنامج إلى بعض المناطق من الذاكرة . البتة 12 (C) تشير إلى طريقة العمل في التحكم C=0 تدل على طريقة العمل BC . البتة 14 (W) ، تساوي 1 عندما تكون الوحدة المركزية غير فعالة ، في حالة الإنتظار (Wait) .

- البتة 15 (P) تعادل 1 عندما تكون الوحدة المركزية في الصيغة مسألة ، والتعليقات المميزة هي أيضاً متنوعة . وتعادل هذه البتة صفراً في صيغة العمل (Supervisor) أي المشرف .

- كود الإنقطاع : عندما يحدث أي إنقطاع ، فإن الكلمة القديمة PSW للبرنامج المقطوع تُخزن في الذاكرة ويوجد فيها كود خاص يُعرف عن طبيعة الإنقطاع . ILC (البتتان 32 و33) (Instruction Length code) . عند حدوث إنقطاع نجد في هاتين البتتين طول آخر تعليمة جرى تفسيرها .

- CC (البتتان 34 و35) . عبارة عن الكود - الشرط الذي يعطي نتيجة المقارنة ، إشارة التأثير بعد تعليقات عديدة ...

- عنوان التعليمة (البتات من 40 إلى 63) . هو عبارة عن عنوان التعليمة التالية المطلوب تنفيذها . تعرف PSW في لحظة الإنقطاع ، هذا الحقل يدل إذن على عنوان التعليمة حيث يجب أن يُعاود البرنامج عمله .

2.3.4 . الكلمة PSW في صيغة العمل EC (البتة 12=1) . تختلف عن السابقة بواسطة إلغاء أتمة القنوات ، وكود الإنقطاع والكود ILC . ويستبدل ذلك بواسطة قناع «R» يدعى «program event recording mask» وبتة T

5 لغة الآلة

1.5 . نسق التعليمات الآلية

لقد أدت بنا دراسة المكنة البسيطة إلى تعريف التعليمات الآلية بطول ثابت ، والمركبة من كود للعملية ومن حقل للعنوان . تهتم العملية بتأثير واحد ، بينما يكون التأثير الثاني موجوداً في مصرف الشحن أو المروم (Accumulator) .

تتمتع المكنات IBM 360/370 بأولية للعنونة أكثر تعقيداً ، تستعمل عدة مرافق وتتمتع بـ 16 مصرفاً عاماً يمكن أن تستعمل كمراصف شحن . نرى إذاً أن تعليمة بعنوان واحد ستكون مركبة من :

- كود للعملية (op. code) .
- رقم مصرف الشحن المعتمد في التعليمة .
- القسم عنوان الذي يتألف من :
- رقم المرفق القاعدي ،
- رقم المرفق النليطي (المؤشر) إذا كان مستعملاً ،
- قيمة الإزاحة .

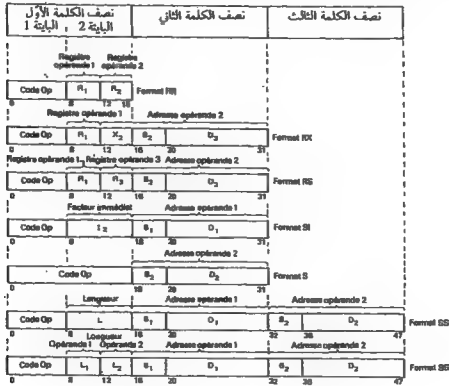
سيتم شرح تعليمات المكنات IBM 360/370 بواسطة ستة أشكال (نسق) مختلفة تتعلق بطبيعة التأثيرات . التعليمات ذات النسق RR (Register to Register) لا تستعمل سوى مصرفين . التعليمات من نوع RX تعالج عدداً موجوداً في أحد المرافف وآخر على عنوان معين في الذاكرة وهذا العنوان يمكن أن يكون دللياً أو مؤشراً . النسق RS (Register and Storage) ، و SI (Storage Immediate) ، و SS (Storage and Storage) ، و لا تسمح بأي عملية تأشير..

الجدول التالي يُمجّد نسق التعليمات المستعمل . الحقل R ، X ، B ، D تمثّل على التوالي أرقام المرافف ، المرافف الدللية ، مرافق القاعدة وقيمة الإزاحة . الحرف L يرمز إلى طول التأثير ويُقاس بالبايتة في التعليمات بالنسق SS . الدليلان 1 و 2

يربطان هذه المعلومات بالتأثير الأول والثاني .

سنلاحظ إن البايته الأولى تحتوي دائماً على كود العملية (ما عدا بالنسبة للنسق S الذي يستعمل 2 بايته) ، إن نصفي الكلمة الثاني والثالث هما عبارة عن عناوين بشكل قاعدة وإزاحة . من المهم أن نتذكر أن التعليقات يجب أن تكون محصورة في نصف كلمات .

تتمتع التعليمة من نوع RX التي تستعمل عنواناً غير مؤشر بحيز X2 يعادل الصفر . والتعليمة التي تستعمل عناوين غير مرتكزة على قاعدة سيكون فيها الحيز B صفرأ . وبالتالي : فإن المرصف 0 لا يُستعمل لا كدليل ولا كمرصف قاعدي .



جدول 1.5

كود العملية المبتدئ بـ

الطول بالبايتات

Code opération commençant par	النسق Format	Longueur en octets
00	RR	2
01	RX	4
10	RX أو S , SI , RS	4
11	SS	6

جدول 2.5

وفي النهاية ، يُمكن أن نُذكر بأن البيتين رقم 1 و2 من كود العملية ترمزان إلى طول ونسق التعليمة . الجدول 2.5 يوضح لنا ذلك .

2.5 . فئات التعليقات

من الممكن تصنيف التعليقات الآلية ضمن ست فئات :

1 - تعليقات التبادل :

- من مرصِف إلى مرصِف .
- من الذاكرة إلى مرصِف (شحن المرصِف LOAD) .
- من مرصِف إلى الذاكرة (STORE) .
- من الذاكرة إلى الذاكرة .
- شحن تلقائي لأحد المراصِف .
- شحن تلقائي للذاكرة .

2 - التعليقات الحسابية :

- الجارية على أعداد بالنظام الثنائي البحت (فاصلة ثابتة) ،
- على أعداد بفاصلة متحركة ، بدقّة بسيطة ، بدقّة مزدوجة أو بنسق موسّع ،
- على أعداد بالنظام العشري المُكثَّف ،
- عمليات المقارنة الحسابية .

3 - التعليقات المنطقية :

- التقاطع ، الاتحاد ، المكاملة ...
- المقارنة المنطقية .

4- تعليقات التحكم بتوالي التعليقات (تعديل مضمون عداد البرنامج PC) .

- تفريع إلزامي . .
- تفريع مشروط .

5- تعليقات الإدخال / الإخراج (Input / Output)

6- تعليقات متفرقة :

- تحويل النسق ، إختيار PSW ، الإزاحة ...

هذه التعليقات تعالج كلمات ، نصف كلمات ، كلمات مزدوجة أو سلاسل من السّيات . إضافة لذلك نجد عدة تعليقات للجمع حسب طول المتأثرات ، ومواقعها في الذاكرة أو في المراصِف ، أو حسب تكوينها الداخلي . مجموع التعليقات يتجاوز إذاً 150 تعليمة .

3.5 . كتابة البرنامج بلغة الآلة

هدف هذا المثل هو الإعتماد على نسق التعليقات الآلية . نقترح جمع مضمون

كلمتين وخزن النتيجة في الذاكرة .

كما ذكرنا أعلاه ، فإن جميع العناوين تُحسب بالنسبة إلى قاعدة (أساس) . إلمم الأول للمبرمج هو في حفظ واحد من 15 مرصفاً عاماً كمرصف قاعدي . نختار مثلاً المرصف رقم 15 .

هكذا ، فإن جميع التعليمات التي تستعمل عناوين مستحتوي على «F» في الحقل المحفوظ للقاعدة .

كتابة البرنامج بلغة الآلة يتطلب اختياراً جيداً لعناوين وجود أو إدخال المعلومات في الذاكرة والمناطق المؤقتة لحفظ النتائج .

تسمح لنا أوالية العنوان القاعدية والإزاحة بعلم الاهتمام بالعنوان الفعلي للمعلومات في الذاكرة . نعتمد في تفكيرنا العناوين النسبية . لنفترض إذاً أن المتأثر الأول موجود على العنوان 0 والثاني في الكلمة التالية ، أي بدءاً من البايته رقم 4 . لنختر الكلمة الثالثة لتخزين النتيجة . ولنفترض أيضاً أن المتأثر الأول يعادل 29 والثاني يعادل 3- . فلنجعل حيز النتيجة صفراً في البداية . وكي نستطيع تمثيل مضمون حيزات الذاكرة يجب علينا أيضاً تحديد طريقة التمثيل المعتمدة للأعداد . ولنختر الأسهل ، صيغة الأعداد بفاصلة ثابتة . حيز المعطيات في برنامجنا هو إذاً ممثّل بالنظام السادس عشري على الطريقة التالية قبل تنفيذ البرنامج :

المتأثر الأول	المتأثر الثاني	المتأثر الثالث
0 0 0 0 0 0 1 0	F F F F F F F 0	0 0 0 0 0 0 0 0
0	4	8 12

من الممكن تصوّر ثلاثة حلول مختلفة لكتابة برنامجنا :

الحل الأول

شحن (LOAD) المتأثر الأول في مرصف نعتبره لاحقاً مرصفاً للشحن من نوع Accumulator (يتم ذلك بواسطة تعليمة من نوع RX بين المرصف والذاكرة) ، جمع المتأثر الثاني إلى هذا المرصف (تعليمة RX) ، وخزن مضمون المرصف في حيز النتائج (تعليمة من نوع RX) .

لنختر المرصف 2 كمرصف للشحن (مركم) . كود عملية التعليمة الشحن (أنظر الملحق) هو 58 ، والتعليمة تكتب بالنظام السادس عشري :

- حيز كود العملية (COP) 58
- الحيز R1 2 (مرصف الشحن)

- حيز الدليل (index) 0 (بدون تأثير)
- مصرف القاعدة F (المصرف 15)
- الإزاحة 0

أي :

5	R	2	0	F	0	0	0
COP	R_1	X_2	B_2	D_2			

تمثل المعطيات بفاصلة ثابتة ، مستعمل التعليمة بكود العملية 5A التي تؤمن جمع مضمون الخلية ذات العنوان $B_1 + X_2 + D_2$ إلى المصرف المذكور في الحيز R_1 أي :

5	A	2	0	F	0	0	4
COP	R_1	X_2	B_2	D_2			

004 = إزاحة المتأثر الثاني بالنسبة إلى القاعدة .

وفي النهاية ، سنخزن النتيجة (التعليمة STORE ، بالكود 50) في الكلمة الثالثة على العنوان 8 .

5	0	2	0	F	0	0	8
COP	R_1	X_2	B_2	D_2			

بإمكاننا أن نفحص صورة البرنامج بعد تخزينه في الذاكرة .

العناوين الموجودة هنا هي العناوين النسبية ولا تتأثر بالعنوان الفعلي لموقع تخزين البرنامج . عنوان الاطلاق في التنفيذ ، أي عنوان أول تعليمة للتنفيذ ، هو عنوان القاعدة + C .

0	0	0	0	0	0	0	1	D
4	F	F	F	F	F	F	F	D
8	0	0	0	0	0	0	0	0
C	5	8	2	0	F	0	0	0
10	5	A	2	0	F	0	0	4
14	5	0	2	0	F	0	0	8
18								

عنوان الاطلاق
في التنفيذ

الحل الثاني :

إنشحن المتأثرين الأول والثاني في المرافف ، وقم بعملية جمع لمضمون مرصف مع المرصف الآخر ومن ثم خزن النتيجة . نستعمل المرافف 2 و 3 كمرافف للعمل والمرصف رقم 15 كمرصف قاعدي . والبرنامج هو التالي :

0	0	0	0	0	0	1	D	
4	F	F	F	F	F	F	D	
8	0	0	0	0	0	0	0	
C	5	8	2	0	F	0	0	R2 شحن المتأثر الأول في
10	5	8	3	0	F	0	4	R3 شحن المتأثر الثاني في
14	1	A	2	3				جمع في R2 .
16	5	0	2	0	F	0	8	خزن النتيجة

هذا الحل يحتاج إلى تعليمة إضافية . متلاحظ وجود تعليمة من نوع RR بطول 2 بايتة .

الحل الثالث :

الحل الثالث كان سيقوم على إجراء الحساب مباشرة في الذاكرة دون استعمال المرافف . وسيحتاج إلى وجود تعليمة بثلاثة عناوين (المتأثر الأول ، المتأثر الثاني والنتيجة) . إلا أن هذا النوع من التعليقات هو غير موجود هنا .

خلاصة

نلاحظ ، في الأمثلة المذكورة ، أن حيز المؤشر (index zone) غير المستعمل هو مصفّر تماماً كما ذكرنا في الفقرة 1.5 .

إن البرمجة بلغة الآلة تبدو معقدة ودقيقة رغم بساطة المثل وعدم إتمامه . لهذا السبب لا نستعمل هذا النوع من البرمجة ونفضل عليه مرونة لغة المؤول (الاسمبلر) .

6 . لغة المؤول ASSEMBLER

المثل البسيط الذي جرى عرضه في الفصل السابق أثبت لنا جميع صعوبات البرمجة بلغة الآلة مع أنه جرى تبسيط كبير لعملنا باستعمال النظام السادس عشري بدلاً من النظام الثنائي .

في لغة الآلة ، فإن أكواد العمليات والعناوين هي رقمية . وكل تعديل في موقع المعطيات يؤدي إلى تعديل العناوين في التعليقات المتعلقة بها .
هذه الصعوبات أدت بالمصممين إلى تعريف لغة ، تُدعى المؤول (assembler) ، قريبة من لغة الآلة ولكنها سهلة الإستعمال مما يجعل ترتيبها في مصاف اللغات المتطورة .

1.6 . مميزات لغات التأويل

- 1 - تتميز التعليقات بلغة المؤول بكود عمليات تذكيري . مثلاً : تعليمة شحن المرصيف من خلال مرصيف آخر تتمتع بكود رمزي هو LR (LOAD TYPE RR) ، وتمتاز تعليقات الجمع بكود رمزي يبدأ بالحرف A ...
- 2 - بإمكان المبرمج أن يقوم بتحديد عناوين بواسطة أسماء رمزية ويقوم برنامج ترجمة المؤول إلى لغة الآلة بربط القيمة الرقمية المناسبة بهذه الأسماء .
- 3 - تتمتع لغة المؤول ليس فقط بمجموعة التعليقات الآلية التي تتضمنها لغة الآلة ، ولكن ببعض التعليقات الخاصة الآلية التي تدعى (التوجيهات) (أو أشباه التعليقات Pseudo-Instruction) وبعض الماكرو تعليمات (macro-instructions) .

2.6 . تعريفات

تدعى تعليمة - آلية كل تعليمة مكتوبة بلغة المؤول ومترجمة إلى تعليمة وإحالة فقط بلغة الآلة . يتناسب كود رقمي مع كود - العملية التذكيري . مثلاً ، عملية نسخ المرصيف 12 في المرصيف 3 ، تكتب بلغة المؤول على الشكل التالي :

وتُترجم إلى لغة الآلة بواسطة :

1	8	3	C
COP	R ₁	R ₂	

يُدعى أمر من نوع توجيه directive كل طلب إلى المؤول ، لا يُؤلد أبداً تعليمة آلية ولكن يُقدم توجيهات للتأويل والتجميع . يوجد نوعان من التوجيهات : تلك التي لا تؤدي إلى أية عملية حجز للذاكرة وتلك المستعملة لحفظ موقع من الذاكرة أو تعريف الثوابت المقيدة للمسألة . هكذا ، فالتعليمة USING#15 تعني إن المرفص 15 سيُعتبر أولاً كمرفص قاعدي ، مما سيسمح بعدم ذكر القاعدة (Base) في التعليقات التالية . هذا التوجيه لا يشغل مكاناً من الذاكرة في الكود المؤلد ، وليس هو سوى إشارة إلى برنامج التأويل والتجميع . أن نكتب 'X'FOFO' DC يعني أن نطلب إلى المؤول حجز بايتين من أجل تخزين الثابتة المحددة بالنظام السادس عشري بواسطة FOFO . لا يوجد توليد لتعليمة ولكن فقط حفظ لمكان من الذاكرة . من الممكن تشبيه هذه التوجيهات بتعليقات التصريح في اللغات المتطورة . أن نكتب بلغة فورتران الأمر dimension TAB (100) يعني أن نطلب من المصترف (Compiler) حفظ المكان من الذاكرة اللازم لاستيعاب الجدول (TAB (100) .

نسُمي ماكرو - تعليمة (MACRO-INSTRUCTION) كل طلب إلى البرنامج المؤول assembler باستبدال سلسلة معروفة مسبقاً من التعليقات تدعى ماكرو - تعريف . الماكرو تعريف هو إذاً عبارة عن مجموعة من التعليقات ينسخها البرنامج assembler مكان كل ماكرو - تعليمة . يقدم النظام مجموعة من الماكرو - تعريفات تدعى نموذجية (ستاندارد) تُسهّل على المبرمج القيام ببعض العمليات المعقدة ، كعمليات الإدخال - الإخراج . كما باستطاعة المبرمج أن يقوم بتعريف نظام خاص به من الماكرو - تعريفات .

3.6 عملية التأويل

الإسم «assembler» يعني في نفس الوقت اللغة والبرنامج الذي يقوم بترجمة النص إلى لغة - الآلة . سنقوم هنا بتناول مرحلة الترجمة بصورة موجزة . يبدو المؤول وكأنه عبارة عن مصترف أو كأنه عبارة عن برنامج لترجمة النص المكتوب بلغة منبع إلى نص مستهدف يتألف من تعليقات - آلية . تدعى عملية الترجمة تأويلاً «assembling» .



1.3.6 . عداد المواقع

يجب على المؤول ، ومن خلال نص منبع ، أن ينتج نصاً ثنائياً يكون مع بعض التحويلات عبارة عن صورة البرنامج المطلوب تنفيذه . لتخصيص عناوين متتالية للتعليقات ، يستعمل المؤول عدداً للمواقع نرمز إليه بواسطة CE . في بداية عملية التاويل فإن CE يبقى ، مثلاً بصفر . وخلال ترجمة التعليقات فإنه يزيد من قيمته حسب طول التعليقات المترجمة . وعندما يلتقي توجيهاً من نوع حجز لموقع أو منطقة من الذاكرة ، فإن مضمون CE يزداد حسب طول المنطقة المحجوزة . كل توجيه من نوع إشارة إلى برنامج التاويل لا يؤدي إلى زيادة في مضمون CE لعدم توليد أية تعليمة آلية . التعليقات ذات النسق RR تؤدي إلى زيادة مضمون CE 2 بايتة ، أما تلك التي تتمتع بنسق RS ، SX ، و S١ فتؤدي إلى زيادة أربع بايتات إلى مضمون CE ، أما تلك ذات النسق SS فتؤدي إلى زيادة 6 إلى مضمونه . وكل توجيه لحجز كلمتين من الذاكرة يؤدي إلى زيادة مضمونه 8 بايتات .

في المثل التالي ، STARTO هي عبارة عن توجيه يؤدي إلى خيئة CE وتصغيره . لا يحدث أي توليد لتعليقات جديدة وبالتالي فإن CE يبقى صفراً . STM 14, 12, 12(13) هي عبارة عن تعليمة من نوع RS تؤدي إلى زيادة 4 إلى مضمون CE . والتوجيه DS 1F يؤدي إلى حفظ كلمة من الذاكرة يرمز إليها بواسطة ALPHA . و CE تزداد قيمته 4 بايتات . التعليمة LR 0,1 بالنسق RR تجعل مضمون CE يزداد 2 .

ملاحظات	منطقة التأثيرات كو- العملية	الرمز الرمز	CE النظام الساس عشري
تصغير CE تعليمة من نوع RS	START 0 STM 14,12,12(13)		0 0 4
حجز كلمة	DS 1F	ALPHA	20 24
تعليمة من نوع RR	LR 0,1	DEBUT	48 4A

وبالاختصار ، فإن عدد المواقع هو عبارة عن كلمة - ذاكرة يُخزَّن فيها المؤول :
 قبل تأويل التعليمة ، عنوان بداية التعليمة (المتعلِّق بتهيئة CE) ،
 - بعد التأويل ، عنوان الخلية الأولى المتوفرة .
 من الممكن أن نلاحظ إن قيمة CE تعادل قيمة مضمون عداد البرنامج عند التنفيذ .

2.3.6 . العنوان الرمزية والمرجعيات المطلقة

لقد ذكرنا سابقاً أنَّ أحد أهم مميزات وخصائص المؤول تكمن في إمكان تسمية العناوين والقيم بواسطة رموز . يمكن أن يكون الرمز عبارة عن إسم منطقة من الذاكرة . في الجدول السابق ، فإنَّ ALPHA و DEBUT هما عبارة عن عنوانين رمزيين نستطيع بلوغهما والعودة إليهما . سيكون بإمكان المبرمج أن يراجع مناطق من الذاكرة تبعاً لذين العناوين بواسطة تعابير من نوع $ALPHA + 8$ ، $DEBUT - 2$.

يُستعمل الرمز * لتسمية القيمة التي يأخذها CE في لحظة التأويل ، أي عنوان البايته اليسرى من التعليمة الموجودة في طور التأويل . من الممكن أن نعود أيضاً بواسطة 2- * إلى العنوان الجاري ناقص 2 بايئة .

ستلاحظ أيضاً أنه لا يمكن لقيمتين مختلفتين لمضمون CE أن يحملتا نفس الإسم . إذ نكون عندئذٍ في حالة التعريف المزدوج .

يسمح المؤول أيضاً ببلوغ قيم مطلقة بواسطة رموز ، أي رموز غير متغيرة عند ترجمة البرنامج . تكتب عملية نسخ المرصف 1 في المرصف 0 مثلاً : $LR\ 0,1$.
 يمكننا أيضاً أن نكتب ، بشكل أوضح $LR\ R0, R1$ بشرط تحديد كون $R0$ و $R1$ عبارة عن رمزين مطلقين يعادلان القيمتين 0 و 1 .

وفي النتيجة ، فإن المؤول سيربط بكل رمز قيمة تدعى قيمة - خاصة ، وهذه القيمة سيتم ترجمتها أو علمه حسب الحالة .

3.3.6 . جدول الرموز

عند العمل ، وفي كل مرة يلتقي المؤول رمزاً معيناً في منطقة الوسم (Label) يقوم بتخصيص خاصيات له :

- خاصة - قيمة تعادل قيمة CE في هذا الموقع .
 - خاصة - طول تعادل البعد (الحجم) بالبايتات للمنطقة المعينة .
- يمكن أن يقوم المؤول إذاً ببناء جدول من الرموز على الشكل التالي :

خاصية - طول	خاصية - قيمة	وسم رمزي
4	20	ALPHA
...	...	BETA
2	48	DÉBUT
...

عندما يلتقي رمزاً معيناً في قسم العنوان من التعليمة ، يقوم المؤول باستشارة هذا الجدول . فإذا كان هذا الرمز موجوداً فيه معني ذلك أنّ الرمز محدد مسبقاً ، وإلا فذلك يعني مرجعاً إلى الأمام ، أي إنه لم يلتقي الرمز حتى الآن في منطقة الوسم ولكنه سيكون لاحقاً (إلا إذا كان يتعلق ذلك برمز خارجي ، أنظر الفصيلين 20 و 21) .

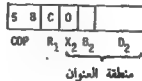
4.3.6 . تأويل التعليمة

يتعلّق ذلك باختيار كيفية ترجمة التعليمة بواسطة المؤول وبالأخص كيف يقوم بتحويل العنوان الرمزي الى عنوان قاعدي ، مؤشر وإزاحة . سنقوم بتحليل ذلك من خلال مثل معين .

لنفترض التعليمة التالية :

L 12, ALPHA
المحمل الثاني
المحمل الأول

إنّها تعليمة من نوع RX ويكود عملية 58 (أنظر الملحق) حيث معناها هو « شحن مضمون الخلية ذات العنوان ALPHA في المرصف رقم 12 » . يقوم عمل المؤول على تعبئة مختلف حقول التعليمة بالنسق RX ، أو :



فلنلاحظ منذ الآن إن منطقة الدليل هي صفر ، لأنه لم يذكر أي مرصف مؤشر أو دليل في العايل الثاني من التعليمة (الحقل الثاني منها) . ولتكتملة حيز العنوان - يجب :

- معرفة المرصف المستعمل كقاعدة .
- معرفة إزاحة العنوان ALPHA بالنسبة للعنوان القاعدي .

ونشير إلى أن العنوان القاعدي لا يختلط بالضرورة مع عنوان وجود البرنامج في الذاكرة .

سنقوم باقتراض في المثل إن ALPHA تناسب القيمة 1C للعداد CE ، وإن المرصف 15 هو مرصف القاعدة وإن العنوان القاعدي يناسب القيمة C للعداد CE .
إزاحة ALPHA بالنسبة إلى القاعدة هي إذا C-C 1 أي 10 .
التعليمة الآلية المؤولة ستكون إذا :

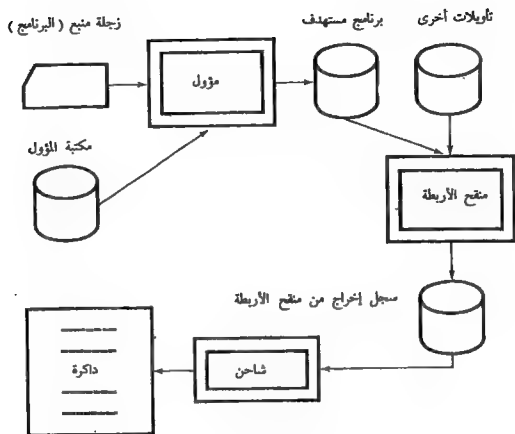
5	8	C	0	F	0	1	0
---	---	---	---	---	---	---	---

4.6 . مراحل تنفيذ البرنامج

إن تنفيذ البرنامج المكتوب بلغة المؤول ، كما بالنسبة للبرنامج المكتوب بإحدى اللغات المتطورة ، يتطلب عدة مراحل . المرحلة الأولى هي مرحلة التأويل والتجميع التي تكلمنا عنها . يُترجم النص الأولي إلى لغة الآلة ويُسخّر في سجل على الأسطوانة المغناطيسية . المرحلة الثانية ، التي يمكن أن تكون اختيارية للبرامج البسيطة ، هي تنقيح الأربطة (link editor) . وتؤدي إلى إجراء بعض الوصلات بين مختلف الزجل المؤولة بشكل منفصل أو التي تشكل جزءاً من مكتبة البرامج . منقح الأربطة يُشكل زجلة واحدة مستهدفة ، يمكن أن تتمتع بهيكلية تغطية ، من خلال مختلف عمليات التأويل . المرحلة التالية تقوم على شحن الزجلة في الذاكرة ، أي إعطائها عنواناً فعلياً لحزنها . وفي هذه الحالة تكون العناوين القاعدية متجمّدة ، وبعض المعلومات المتعلقة بالعناوين المطلقة يجب أن تُحسب من جديد . يكفي إذاً أن نقوم بتخزين عنوان أول تعليمة للتنفيذ في عداد البرنامج CO (للكلمة الثانية من PSW) للبدء بمرحلة التنفيذ .

سنسمي نقطة الشحن أو عنوان الحزن ، عنوان بداية المنطقة المُخصّصة للبرنامج . سيُدعى عنوان الإطلاق عنوان أول تعليمة للتنفيذ من البرنامج . نقاط الدخول إلى البرنامج هي عناوين ، التعليقات أو المعطيات ، من الممكن بلوغها من خارج البرنامج . تتصل نقاط الدخول هذه بمنقح الأربطة الذي يمكن أن يقوم بإجراء وصلات بين مختلف الزجل (modules) . عنوان الإطلاق هو نقطة دخول .

بدون إعطاء جميع الإمكانات فإن المخطط 1.6 يعرض مختلف المراحل الواجب أن يتبعها البرنامج كي يجري تنفيذه .



شكل 1.6

القسم الثاني

المؤول 360/370

7 . العناصر الأساسية

1.7 . عموميات وتقديم البرنامج

- 1 - مجموعة السيات :
يستعمل المؤول السيات الأبجدية A ، B ، Z... ، @ ، \$ ، والأرقام 0 ، 1 ، 2 ... 9 ، والسيات الخاصة : + ، * ، / ، = ، (،) ، ' ، & والقسمة البيضاء . (blank)

- 2 - ورقة البرنامج
المنطقة المحجوزة للمؤول تمتد من العمود 1 إلى العمود 71 . المنطقة 73 إلى 80 لا تُفسّر من جانب المؤول وتُستعمل لتحريف التعليقات . العمود 72 يُستعمل عندما ترغب إحدى التعليقات بالمتابعة على السطر التالي . تقسّم منطقة التعليمة (1 إلى 71) إلى أربعة أقسام :
منطقة الرموز : وتُستعمل لاجراء تخصيص رمزي للتعليمة (وسم) أو إلى معطى (اسم المعطى) .

الاسم المُخصّص :

- يبدأ بالعمود 1 بواسطة سمة أبجدية .
 - يحتوي على أكثر من 8 سيّات أبجدية .
 - لا يحتوي على فراغ أو سيّات خاصة .
- الرموز التي تظهر في منطقة التأثيرات تخضع لنفس القواعد :

أمثلة :

غير صالح	صالح
(9 سيات) (فراغ) (تبدأ برقم) (تحتوي على سمة خاصة)	RESULTATS TAB 1 1ABC BC-1
	A1234567 ZONE @123 ### \$ABC

منطقة العملية : وتستعمل لتحديد كود - العملية الخاص بالتعليمة . هذا الحيز يبدأ في أي مكان ، إنطلاقاً من العمود رقم 2 . إلا أنه يجب أن ينفصل الرمز عن كود العملية بواسطة فراغ واحد على الأقل .

منطقة العوامل (العناوين) : وتحتوي على العناوين أو على المتأثرات . تبدأ هذه المنطقة من أي عمود على يمين كود - العملية وتنفصل عنه بواسطة فراغ واحد على الأقل . ويمكن أن تحتوي هذه المنطقة على العناوين ، ولا يمكن أن تحتوي على فراغات وكل عنوان ينفصل عن الآخر بواسطة فاصلة .

منطقة الملاحظات : وتبدأ من يمين أول فراغ يتلو منطقة العوامل وتمتد حتى 71 عموداً . يمكن اعتبار السطر بكامله كملاحظة فيما لو بدأ هذا السطر بنجمة (*) على العمود الأول .

سطر التكملة : كل سمة عدا الفراغ في العمود 72 تشير إلى أن التعليمة الجارية لم تنته وتستابع على السطر التالي . يفترض المؤول أن السطر التالي يبدأ بالعمود رقم 16 ، وبالنسبة فإن التعليمة ستتابع بدءاً من العمود رقم 16 . يسمح بسطرين فقط لتكملة التعليمة .

الحصر العادي : من المفيد حصر هذه المناطق انطلاقاً من الأعمدة 1 ، 10 ، 16 و 40 . ونشير إلى أن الحيز المُفسّر بواسطة المؤول يمتد إلزامياً من 1 إلى 71 وإن الأسطر التابعة تبدأ من العمود رقم 16 . هذه القيم هي قابلة للتعديل بواسطة الأمر ICTL

منطقة الرموز	منطقة الكود العملية	منطقة العوامل	منطقة الملاحظة	منطقة المرئف
1	10	16	40	72 80
ALPHA	DC	C'ABCD'	عمود تابع colonne suite	
	LR	1,2		
" CETTE LIGNE EST UN COMMENTAIRE			(سطر ملاحظة)	
BETA	DC	C'TEXTE	SE CONTINUER	(نصن يتبع)
		T SUR LA LIGNE SUIVANTE	(على السطر التالي)	

جدول 1.7

2.7 . عناصر لغة المؤول

لقد لاحظنا حتى الآن إن المؤول يسمح لنا باستعمال رموز معينة لتسمية العناوين أو القيم . وعملياً فإن لغة للمؤول تسمح لنا :

- باستعمال كتابات مثل '1011' B ، 'A10C' X والتي ستعامل وكأنها قيم باللغة الثنائية ، أو السادس عشرية . . . وهي ستكون عبارة عن القيم المعرفة أوتوماتيكياً .

- بلوغ الطول المتعلق بأحد الرموز . لو افترضنا إن «BIDON» هو رسم تعليمية ، أو بشكل عام ، أكثر اسم حيز معين ، فإن L'BIDON سيحدد طول التعليمية أو المنطقة . ويتعلق ذلك بالخاصية - طول ؛

- استعمال الأحرف كمتأثرات في التعليقات ؛

- خلط كل هذه الإمكانيات لنحصل على تعابير ستكون معادلة لعناوين قابلة للنقل إلى قيم مطلقة .

من الملائم إذاً تحديد القواعد النحوية التي تسمح باستعمال هذه الإمكانيات

1.2.7 . قيم المَعْرِفات الأوتوماتيكية (Auto-definition)

قيمة المَعْرِف الأوتوماتيكي هي واحد من أشكال الكتابة ، معروف من قبل المؤول ، يسمح بتحديد القيمة .

مثلاً :

'B'X ، '1011'B و 11 هي عبارة عن ثلاث كتابات مختلفة تسمح بتحديد القيمة 11 (عشري) الممثلة في المكنة بواسطة تشكيلة البتات 1011 . هذا الشكل في الكتابة هو مسموح ، مع بعض التحديدات ، بداخل حيز العوامل (منطقة العنوان) من التعليمية .

هناك أربعة أنواع من المَعْرِفات الأوتوماتيكية المقبولة :

- الثنائي : '1001101'B وعلى الأكثر 32 رقماً ثنائياً تحت إشراف النظام OS و 24 بالنظام (DOS) .

- السادس عشري : '1A3BC'X ،

- العشري : 125 (حد أقصى 10 أرقام عشرية) .

- نوع السهات : 'A' ، 'C' (سمة أبوستروف أو الفاصلة العليا) ، 'C'ABCD' ، 'C'AB' . يجب أن نحصل كحد أقصى على أربع سهات بالنظام OS وثلاث بالنظام

DOS .

وبشكل عام ، فإن قيمة التعريف الأوتوماتيكي يجب أن تتم على 24 بتة بإشراف النظام DOS وعلى 32 بتة كحد أقصى بإشراف النظام OS . سنجد أمثلة على طرق استعمالها في الفقرة 3.7 المتعلقة بالتعابير .

2.2.7 . التأثيرات الحرفية

- هي عبارة عن قيم مستعملة كمثاثرات في حيز عوامل التعليمات .
لشحن القيمة 125 في المرصف 3 يمكن للمبرمج أن يختار أحد حلين :
- 1- حجز حيز من الذاكرة ، يدعى ALPHA مثلاً ، ويُعرّف عنه وكأنه يحتوي على القيمة 125 ، وبعد ذلك يُشحن ALPHA في المرصف 3 بواسطة التعليمات :
$$L3, ALPHA;$$
 - 2- كتابة التعليمات : $F'125 = L3$ ، وسيهتم المؤول بحجز الحلية من الذاكرة التي تحتوي على 125 في منطقة تدعى POOL (حوض) .
في المثل المذكور لاحقاً ، فإن القارئ سيتحقق :
- من أن المؤول سيضع عنوان المتأثر الحرفي بشكل قاعبة وإزاحة داخل كود التعليمات المؤلد عنه ،
- من أن إستعمالين مختلفين لنفس المتأثر الحرفي لن يؤديا سوى إلى حجز واحد في الذاكرة ،
- من أن للمتأثر الحرفي هو شبيه برمز قابل للترجمة .
إن استعمال المتأثر الحرفي ، إن لم يعمل أي شيء جديد ، فإنه يُقدم لنا فائدة بالنسبة لوضوح كتابة التعليمات .
- قواعد الكتابة
- يُجند المتأثر الحرفي وكأنه متأثر عادي في توجيه DC مسبوق بالإشارة « = » . أما القواعد المتعلقة بمتاثرات التوجيه DC فإنها ستوضح لاحقاً .
- لا يمكن أن يُستعمل المتأثر الحرفي كمعامل في التعبير (فقرة 3.7) الرقمي أو غير الرقمي .
- من البليبي ، لأن المتأثر الحرفي يُستعمل « كمعطى للإدخال » في التعليمات ، أن لا يظهر في الحقل المُستقبل من التعليمات . سيكون من المتنافر أن نكتب : $ST3, F'125$ ($ST =$ خزّن مضمون المرصف في الذاكرة) .

L'OC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000000				1	CSECT
				2	EXTRN SP1
				3	USING
				4	1=0
000000	5810 F018	00017	00000	5	2=C'ABCD'
000000	5820 F01C	0001C		6	2=F'0'
000000	5820 F018	00018		7	2=V(SP)
000000	5820 F020	00020		8	3=C'ABCD'
000000	5830 F01C	0001C		9	1=A(SP)
000000	5810 F024	00024		10	END
000000	00000000			11	=F'0'
000000	C1C2C3C4			12	=C'ABCD'
000000	00000000			13	=V(SP)
000000	00000000			14	=A(SP)

3.2.7 . الخاصية - طول

وتسمح ببلوغ الطول المرتبط بالرمز . ويكتب :

مثلاً : L' symbolic name اسم رمزي ' L

L'ZONE L'SUITE L'*

- إذا كان الرمز هو اسم الحيز ، فهو يأخذ كقيمة طول الحيز مُقاساً بالبايتة .

- إذا كان الرمز هو اسم التعليمة ، فهو يأخذ واحدة من القيم 2 ، 4 أو 6 حسب نوع التعليمة .

- إذا كان الرمز هو (*) ، فهو يأخذ كقيمة طول التعليمة التي يظهر فيها .

بالنسبة للتوجيهين DC وDS . فإن الخاصية - طول لا تتأثر بوجود عامل الإزداجية . ملاحظ أنه بالنسبة للتوجيه EQU فإن قيمة الخاصية - طول هي قيمة المتأثر الأيسر .

الأمثلة التالية ، وللفهم الكامل ، تتطلب بأن نكون أكثر تقدماً في هذه الدراسة .
إلا أننا نعرضها هنا :

الرمز	كود - العملية	عوامل	خاصية -	قيمة
ZONE1	DS	C180	L'ZONE1	80
ZONE2	DS	C1200	L'ZONE2	200
CARAC	DC	C'ABCDE'	L'CARAC	5
ABSOL1	EQU	ZONE2-ZONE1	L'ABSOL1	200
ABSOL2	EQU	Z6	L'ABSOL2	1
INSTR1	LR	Q,1	L'INSTR1	2
			L'*	2
INSTR2	MVC	ZONE2(L'*) , ZONE1	L'INSTR2	6
			L'*	6
	MVC	ZONE2(L'ZONE2-10) , ZONE1	L'ZONE2	200
ALPHA	DC	6F'0'	L'ALPHA	4

3.7 التعابير

تعريف :

التعبير هو تركيب من الرموز ، وقيم التعريف - الأوتوماتيكي وخصائص - الطول في منطقة التأثيرات من التعليمة .

الاستعمال :

تستعمل التعابير لتحديد :

- العنوان ،
- الطول الواضح ،
- المعدل ،
- عامل التكرار .
- التأثير .

فئات التعابير

التعابير هي بسيطة أو مركبة ، مطلقة أو قابلة للترجمة - التعبير البسيط هو الرمز الوحيد أو الرمز * (قيمة عدّاد المواقع عند تأويل التعليمة ، فقرة 1.3.6) .
التعبير المركّب هو مجموعة من عدة تعابير بسيطة مرتبطة بمؤثرات من نوع + ، - ، * ، (1) أو / ، التي تمثّل على التوالي الجمع ، الطرح ، الضرب والقسمة .
أمثلة :

ALPHA+2	++3	*	=	CE
ALPHA-BETA	+~2	*	=	CE
3*DELTA	A*3	*	=	مؤثر
(ALPHA-BETA)/2	++2			تعبير غير صالح
ALPHA+X'1A'	+3			تعبير غير صالح
TAB+L'LIGNE				

قواعد الإنشاء

التعبير المركّب :

- لا يمكن أن يبدأ بمؤثر ،
- لا يمكن أن يحتوي على مؤثرين ثنائيين متتاليين ،

(1) يجب عدم الخلط بين المؤثر * والرمز الذي يمثل عدد المواقع .

- لا يمكن أن يحتوي على نجمتين ،
 - لا يمكن أن يحتوي على تعبيرين بسيطين يتتابعان بدون مؤثر بينهما ،
 - لا يمكن أن يحتوي على متأثر حرفي .
- النظام OS يسمح باستعمال 19 مؤثراً أحادياً وثنائياً و6 مستويات من الأهلة . بينما النظام DOS لا يسمح سوى بـ 15 مؤثراً و5 مستويات .

تقييم التعابير

يقوم المؤول بتخصيص قيمة رقمية لكل تعبير بسيط وبعد ذلك يُقِيم من اليسار إلى اليمين التعبير حسب أولوية خاصة للضرب وللقسمة بالنسبة للجمع والطرح .
 $A+B * C$ تُقِيم وكأنها $A+(B * C)$ وليس كأنها $(A+B) * C$. النتيجة الحسابية تصبح قيمة التعبير ، والمؤول يُقِيم بشكل طبيعي في المكان الأول المؤثرات الأحادية وداخل الأهلة . القسمة على صفر هي صحيحة وتعطي نتيجة صفر .

تعايير مطلقة ، تعابير منقولة

التعبير المنقول هو تعبير حيث القيمة تتغير مقدار n إذا كان البرنامج منقولاً إلى n

بايتة

التعبير المطلق هو التعبير الذي لا تتغير قيمته عند النقل .

أمثلة :

لنفترض إن ALPHA و BETA هي رموز منقولة وإن VAL1 و VAL2 هي رموز مطلقة :

تعايير مطلقة	تعايير منقولة
VAL1+B*101	ALPHA+3
ALPHA-BETA	BETA+L'ZONE
VAL1+VAL2	BETA+VAL1

التعبير سيكون مطلقاً إذا كان يحتوي على :

- رموز مطلقة ، قيم تعريفات أوتوماتيكية ، خاصيات - طول ،
 - رموز منقولة يظهر كل اثنين منها على حدة وتؤدي إلى تصغير فاعلية النقل .
- سنلاحظ إنه إذا كان $T1$ و $T2$ تعبيرين منقولين ، فإن $T1+T2$ و $T1*3$ ليست لا مطلقة ولا منقولة .

ولنتأكد من ذلك يكفي أن نقوم بإجراء عملية نقل ؟؟ دار 100 مثلاً :

$T1 + 100$	$T1$ تصبح
$T2 + 100$	$T2$ تصبح
$T1 + T2 + 100$	$T1 + T2$ تصبح
$T1 * 3 + 300$	$T1 * 3$ تصبح

التعابير لا تحتل نفس الإزاحة .

إستعمال التعابير هو بشكل خاص مفيد لأنه يسمح بتحديد العناصر حيث القيم هي قابلة للتغيير عند التأويل وذلك بشكل مُعَامِلَات ومتغيرات (مثلاً صفحة 122 ، السطر 78 من البرنامج) . كل تعديل في قيمة المتغير من التعبير سيكون محسوباً من جديد بواسطة المؤول وليس بواسطة المبرمج ، مما يُسهّل عمل المبرمج .

8 توجيهات تعريف الرموز

لنأخذ هذه القطعة من برنامج بلغة فورتران :

```

DIMENSION TAB(100)
DO 50 I=1,100
  TAB(I)=1
50
-----

```

يطلب الأمر DIMENSION حجز 100 كلمة - ذاكرة مجموعة تحت اسم الجدول TAB . تدل القواعد الضمنية المتعلقة بنوع المَعْرِفَات أنَّ هذا الجدول سيتألف من أعداد حقيقية ، أي مَكُونَة في التمثيل بفاصلة متحركة بدقة بسيطة . يعرف المَصْرُف بأنه يجب أن يستعمل ، لتوليد كود التعليقات الحسابية التي تبلغ TAB ، التعليقات الحسابية بدقة بسيطة .

وفي فورتران ، كما في جميع لغات البرمجة ، كل رجوع إلى مَعْرِف يفترض أن يكون الأخير معروفاً من المَصْرُف ، أي مُحدَّداً خلال البرنامج بواسطة نوعه (حقيقي ، صحيح ..) وطوله مُقاساً بالكلمات أو بالبايتات . وفي النهاية يُنصَّب المَعْرِف TAB بخاصية - قيمة (قيمة المَعْرِف ستكون عنوانه) وبخاصية - طول (بعد الحيزِ المُشار إليه بالبايتة) .

في لغة التاويل المسألة هي نفسها ، يجب أن يُحدَّد كل رمز بواسطة خواصه . سنرى توجيهين DC وDS يسمحان بتعريف الثوابت وحجز مكان من الذاكرة ، والتوجيه EQU الذي يسمح بإجراء توازنات بين الرموز .

1.8 . تعريف الثابتة DC

كثير الإستعمال ، هذا التوجيه يسمح بحجز منطقة من الذاكرة تحتوي على القيمة المدعومة ثابتة ويسميتها بواسطة أحد الرموز .

شكل هذا التوجيه هو التالي :

رمز	كود العملية	عامل
[وسم]	DC	$d t m 'c'$

- الوسم هو الإسم الرمزي للثابتة وهو اختياري .
 d - هو عامل الازدواجية ، وهو اختياري ، وإذا كان مهماً فلن قيمته تعادل 1 . إنه يشير إلى العدد الذي يجب أن تولد فيه الثابتة .
 t - هو النوع ، يمكن أن يكون أحد الأكواد الموجودة في الجدول التالي :

الاصطفا	الطول الضمي	نسق المكنة	نوع الثابتة	كود
بايتة		EBCDIC	سمة	C
بايتة		ثنائي فاصلة ثابتة	سامن عشري	X
بايتة		ثنائي	ثنائي	II
كلمة	كلمة واحدة	كلمة ثنائية بفاصلة ثابتة	عشري	F
نصف كلمة	نصف كلمة	نصف كلمة بفاصلة ثابتة	عشري	H
كلمة	كلمة واحدة	فاصلة متحركة ودقة بسيطة	عشري	E
: كلمة مزدوجة	كلمتان	فاصلة متحركة ودقة مضاعفة	عشري	D
: كلمة مزدوجة	4 كلمت	فاصلة متحركة ودقة رباعية	عشري	L
بايتة		عشري موسع	عشري	II
بايتة		عشري مكثف	عشري	P

جدول 1.8

في المكنة تُحصر الثوابت في حدود البايته ، نصف الكلمة ، الكلمة أو الكلمة المزدوجة حسب نوعها ما عدا في الحالة التي نُحلّد فيها طولها (أو نستعمل معدّلاً للطول) .
 m - هو معدّل طول الثابتة ، ويمكن أن يكون :

- أ - معدّل طول ضمني يكتب على شكل Ln حيث n هو عدد البايتات في التمثيل الداخلي . إن وجود معدّل للطول يُصغّر قاعدة الاصطفا الضمنية .
 ب - مُعدّل للحصر يكتب على الشكل التالي : Sn .

معدّل الحصر يقوم بإجراء إزاحة لـ n بته إلى اليسار إذا كانت n إيجابية ، وإلى اليمين إذا كانت n سلبية . أي يقوم بإجراء ضرب أو قسمة صحيحة على 2^n .
 معدّل الحصر ، ويدعى أيضاً المقياس ، يُطبّق على الثوابت E ، D و I .
 'n' هي الثابتة للمحلّدة بين فاصلتين عليّين (') . الثوابت يمكن أن تكون محدّدة بإشارة ، فاصلة عشرية ويأس (قوة) يُرمز إليه بالحرف E . الأمثلة التالية تُظهر لنا مختلف الإمكانيات . وهناك جدول في الملحق يُوجز لنا مميزات الثوابت .

LXC	PROJECT CODE	ADDR1	ADDR2	STAT	SOURCE STATEMENT
000000	00000000				70
000000	00000000				71
000000	00000000				72
000000	00000000				73
000000	00000000				74
000000	00000000				75
000000	00000000				76
000000	00000000				77
000000	00000000				78
000000	00000000				79
000000	00000000				80
000000	00000000				81
000000	00000000				82
000000	00000000				83
000000	00000000				84
000000	00000000				85
000000	00000000				86
000000	00000000				87
000000	00000000				88
000000	00000000				89
000000	00000000				90
000000	00000000				91
000000	00000000				92
000000	00000000				93
000000	00000000				94
000000	00000000				95
000000	00000000				96
000000	00000000				97
000000	00000000				98
000000	00000000				99
000000	00000000				100

2.8 . ثوابت العنوان⁽¹⁾

إنَّ تعريف ثابتة - عنوان يعني حجز مكان من الذاكرة لتخزين عنوان أحد العناصر . نشير هنا إلى بعض المفاهيم الأساسية . العنوان الفعلي ، أي العنوان الحقيقي لأحد العناصر هو غير معروف إلا عند شحن البرنامج في الذاكرة . لذا فمن غير الممكن ، في مرحلة التأويل والتجميع ، أن يكون بتصرفنا العنوان الفعلي الخاص بالرمز . نبلغ الرمز بواسطة الإزاحة نسبة إلى مضمون مصرف القاعدة .

في بعض الأحيان يبدو من غير الممكن بلوغ أحد الرموز التي لا تنتمي إلى الزجلة التي تكون في طور المعالجة من قِبَل المؤول . هذه هي الحالة ، مثلاً ، عندما نرغب بإجراء تفريع إلى برنامج - ثانوي مؤول ومترجم على حدة . الحلّ يقوم إذاً ، بالنسبة للمؤول ، على بلوغ مباشر بسبب وجود كلمة ، تدعى ثابتة - عنوان ، يقوم الشاحن (Loader) بمثلتها بشكل مناسب .

مثلاً :

نرغب ، للتفريع إلى المصرف 15 ، شحنه بعنوان نقطة الدخول P1 لبرنامج - ثانوي مؤول على حدة . سنحفظ ، في الزجلة المتأدية ، كلمة تدعى هنا ADRP1 سيتم تعريفها كثابتة عنوان خارجية . والمؤول سيقوم بإعدادها وتصغيرها ، كما سيقوم الشاحن بتخزين العنوان الفعلي P1 في داخلها . العنوان P1 سنحصل عليه إذاً في المصرف 15 بواسطة التعليمة :

. L 15, ADRP1

إنَّ نسق تعريف ثابتة العنوان هو التالي :

عامل	كود - العملية	رمز
$d + m \text{ (c)}$	DC	[وسم]

نسق هذا الأمر لا يتميز عن نسق تعريف الثوابت إلا بتبديل الفواصل العليا بالأهملّة .

- d هو عامل الإزدواجية ، وإذا جرى إسماله فإنّه يعادل 1 .
- t هو كود نوع الثابتة .

(1) دراسة هذه الفقرة القيمة للفهم الكامل يمكن أن يفقر عنها عند القراءة الأولى .

وقد يكون A ، Y ، S ، V أو Q (النوع Q ليس متوقراً سوى تحت النظام OS) . النوعان A وY يسمحان بتعريف الثوابت بواسطة تعابير بسيطة أو مركبة ، مطلقة أو منقولة . القيمة ثابتة العنوان محلّة لجهة اليمين في كلمة (نوع A) أو نصف كلمة (نوع Y) . الثوابت من نوع S تسمح بتخزين عناوين بشكل قاعدة وإزاحة على نصف كلمة . ولا يمكنها أن تعرف في نص حرفي . تستعمل الثوابت من نوع V لتعريف عناوين خارجية من نوع « إسم برنامج ثانوي » .

- m هو عبارة عن معدّل الطول الضمني . وجود المعدّل يؤدي إلى إلغاء قاعدة الاصطفاف الأوتوماتيكية (alignment) .

- C هو عبارة عن الثابتة نفسها مكتوبة بدأخل أهلة . الأمثلة في الصفحة 75 تعرض وتعرف كل نوع من الثوابت .

استعمال ثابتة العنوان :

- تستعمل :

- لشحن عنوان في مصرف .

- لاجراء وصلات بين البرنامج والبرنامج الثانوي .

وسيتم درزن ذلك في الفصلين 20 و 21 .

3.8 . أمر حجز مواقع مُن الذاكرة

هذا الأمر هو عبارة عن توجيه يسمح بحجز موقع من الذاكرة دون إعداد أوتبيئة مضمونه عند التأويل . هذا الأمر يؤدي إذا إلى زيادة مضمون عداد المواقع . ويسمح بتسمية المناطق المحلّدة وبلوغها رمزياً . النحو ، القريب من نحو التوجيه DC ، هو التالي :

عامل	كود العملية	رمز
d t m	DS	[وسم]

- d مُعامل الازدواجية ، وهو اختياري . وإذا كان صفراً فهو يسمح بزيادة عداد المواقع حتى حدود نصف كلمة ، كلمة أو كلمة مزدوجة حسب نوع t المرتبطة بالمنطقة . هذه الخصوصية تستعمل كثيراً ونوضّحها في الأمثلة والأسئلة . سنشير هنا ، إلى أنه مع وجود عامل ازدواجية يعادل صفراً ، فإن الوسم الموجود في منطقة الرمز هو مخزّن في جدول الرموز .

LOC	OBJECT CODE	ADDR1	ADDR2	SYNT	SOURCE STATEMENT	SYMBOLS
000000				1	CSECT	
000000		00000		2	EXTRN	
000000		00000		3	USING	
000000		00000		4	END	
000000		00000		5	RELOC	
000000		00000		6	DC	
000000		00000		7	DC	
000000		00000		8	DC	
000000		00000		9	DC	
000000		00000		10	DC	
000000		00000		11	DC	
000000		00000		12	DC	
000000		00000		13	DC	
000000		00000		14	DC	
000000		00000		15	DC	
000000		00000		16	DC	
000000		00000		17	DC	
000000		00000		18	DC	
000000		00000		19	DC	
000000		00000		20	DC	
000000		00000		21	DC	
000000		00000		22	DC	
000000		00000		23	DC	
000000		00000		24	DC	
000000		00000		25	DC	
000000		00000		26	DC	
000000		00000		27	DC	
000000		00000		28	DC	
000000		00000		29	DC	
000000		00000		30	DC	
000000		00000		31	DC	
000000		00000		32	DC	
000000		00000		33	DC	
000000		00000		34	DC	
000000		00000		35	DC	
000000		00000		36	DC	
000000		00000		37	DC	
000000		00000		38	DC	
000000		00000		39	DC	
000000		00000		40	DC	
000000		00000		41	DC	
000000		00000		42	DC	
000000		00000		43	DC	
000000		00000		44	DC	
000000		00000		45	DC	
000000		00000		46	DC	
000000		00000		47	DC	
000000		00000		48	DC	
000000		00000		49	DC	
000000		00000		50	DC	
000000		00000		51	DC	
000000		00000		52	DC	
000000		00000		53	DC	
000000		00000		54	DC	
000000		00000		55	DC	
000000		00000		56	DC	
000000		00000		57	DC	
000000		00000		58	DC	
000000		00000		59	DC	
000000		00000		60	DC	
000000		00000		61	DC	
000000		00000		62	DC	
000000		00000		63	DC	
000000		00000		64	DC	
000000		00000		65	DC	
000000		00000		66	DC	
000000		00000		67	DC	
000000		00000		68	DC	
000000		00000		69	DC	
000000		00000		70	DC	
000000		00000		71	DC	
000000		00000		72	DC	
000000		00000		73	DC	
000000		00000		74	DC	
000000		00000		75	DC	
000000		00000		76	DC	
000000		00000		77	DC	
000000		00000		78	DC	
000000		00000		79	DC	
000000		00000		80	DC	
000000		00000		81	DC	
000000		00000		82	DC	
000000		00000		83	DC	
000000		00000		84	DC	
000000		00000		85	DC	
000000		00000		86	DC	
000000		00000		87	DC	
000000		00000		88	DC	
000000		00000		89	DC	
000000		00000		90	DC	
000000		00000		91	DC	
000000		00000		92	DC	
000000		00000		93	DC	
000000		00000		94	DC	
000000		00000		95	DC	
000000		00000		96	DC	
000000		00000		97	DC	
000000		00000		98	DC	
000000		00000		99	DC	
000000		00000		100	DC	

٤ - يُحدّد نوع المنطقة أي بالتحديد كما جرى بالنسبة للأمر DC . وهو إلزامي ويُحدّد التسطير الضمني .

m - هو معدّل الطول ويُكتب Ln ، حيث n هو طول المنطقة بالبايتات . كما بالنسبة للأمر DC فهو اختياري . وجوده يلغي فعل الإصطفاف الضمني . سنشير هنا إلى أن الطول الأقصى للثابتة من نوع سلسلة السيات المُحدّدة في الأمر DC هو 256 بايتة ، وإستعمال النظام OS يسمح به 65535 بايتة .

لتسهيل صيانة البرامج سنستعمل : ETIQ DSOH لتعريف نقاط التفريع .
قدر المستطاع سنفضل إستعمال الأمر DC عن الأمر DS الذي يقوم بإعداد المنطقة بقيمة محايدة ستكون مريئة في عملية DUMP (دلق) .

4.8 . توجيه التعامل EQU

يسمح بتعريف رمز وإعطائه قيمة مطلقة أو عوّلَة ويُكتب على الشكل التالي :

تعبير مطلق أو عوّل	EQU	(Symbol) رمز
--------------------	-----	--------------

سنشير هنا إلى أن وجود الرمز هو إلزامي . لا يمحّز التوجيه أي موقع من الذاكرة ولا يقوم سوى بإنشاء رمز جديد في جدول الرموز . ويمكن أن يكون موجوداً في أي موقع من البرنامج ويُستخدم :

1- لاستعمال أسماء بدلاً من القيم . تجري العادة مثلاً على كتابة :

R0	EQU	0
R1	EQU	1
...
R15	EQU	15

عما يسمح ، منذ البداية ، ببلوغ المرافص بواسطة الأسماء R0 ، R1 ، ... ، R15 . بدلاً من القيم 0 ، 1 ، ... ، 15 . هذا ما يؤدي إلى فائدة ووضوح في العمل ولكن أيضاً إلى إمكانية إيجاد مراجع المرافص بسهولة لأنها ستظهر في جدول الرموز وفي البلوغ التصالي .

2- لتخصيص قيمة جديدة محدّدة داخل البرنامج لرمز معين ، أي معرّف خلال الأهرط السابقة .

RD	EQU	0	
REGD	EQU	90	(رمز مطلق)
DEB	LR	R1,R2	
DEBUT	EQU	DEB	(رمز محوّل)
ZONE	DS	4F	
Z1	EQU	ZONE+12	(تعبير محوّل)

3- لحساب التباير حيث القيمة مجهولة في لحظة الكتابة أو صعوبة الحساب وتخصيص رمز لها .

EXPRES EQU A-(B+C)/5-0
ETIQ EQU *

(قيمةعداد للمواقع)

تمارين

تمرين 1.8 - ولد ، بواسطة تعريف ثابتة مخصصة ، منطقة من الذاكرة بحجم 100 بايتة تحتوي على سلسلة من 100 عدد صحيح طبيعي . نفس السؤال لمنطقة بحجم 100 كلمة .

تمرين 2.8 - عرف حيز من الذاكرة لاستيعاب رقم الضمان الإجتماعي (13 سمة) مع وصف للهيكلة التالية .



وذلك بفحص الخاصية - طول لكل معروف مذكور .

تمرين 3.8 - باستعمال الأمر ORG (فقرة 3.20)، مطلوب تعريف منطقة من الذاكرة يمكن أن تستوعب إما ثمناً (8 أرقام عشرية موسعة) أو كمية (4 أرقام عشرية موسعة) ، أورتقياً (علداً صحيحاً بفاصلة ثابتة) ونصاً من 10 سيات . يتعلّق ذلك بإعادة تعريف من نوع REDEFINES بلغة كوبرول .

9 كتابة العناوين بلغة المؤول

1.9 . قاعدة ضمنية ، قاعدة جلية

في جسم التعليقات الآلية ، فإن العناوين المحولة تكون ممثلة بواسطة مرصف قاعدي ، وإزاحة ومرصف دليل (حالة النسق RX) . عند كتابة التعليقات - الآلية بلغة المؤول مستقوم بإيجاد ثلاثة متأثرات . لقد لاحظنا حتى الآن أنه كان يوجد ستة أنسقة مختلفة للتعليقات الآلية . إضافة لذلك ، وفي لغة المؤول ، فإن كتابة منطقة العوامل (منطقة العناوين والثوابت) ستتغير حسب نسق المكنة .

لنأخذ تعليمة شحن المرصف 3 (LOAD) من خلال مضمون عنوان معين . لنفترض إن المرصف 15 قد جرى إختياره كمرصف قاعدي ، وإن العنوان موضع السؤال هو موجود على مسافة 512 (في القاعدة العاشرة) من العنوان القاعدي وهو مؤشر بواسطة المرصف 5 . التعليمة - الآلية سيكون لها الشكل التالي :

5	8	3	5	F	2	0	0
CDP	R ₁	R ₂	B ₂	B ₂			

سيكون بإمكان المبرمج بلغة المؤول أن يكتب التعليمة على الشكل التالي : (5, 15) , L 3 . القاعدة 15 هي هنا مسجلة بشكل واضح . لا نرى بهذا الشكل الفائدة الرمزية من لغة المؤول .

لنأمن ببساطة أكبر فإن المؤول يسمح بعلم ذكر القاعدة في منطقة العوامل التابعة للتعليمة . يكفي لذلك أن نصرح ، بواسطة التوجيه 15, * USING ، أن التعليقات التالية يجب أن تؤول (تجمع) مع المرصف 15 كقاعدة . الفائدة الأولى هي السهولة بتعديل مرصف القاعدة دون إعادة كتابة جميع التعليقات . كذلك ، فإن الإزاحة ومرصف المؤشر يمكن أن يتم تمثيلها بشكل رمزي عند الحاجة . هكذا ، فلنأخذ العنوان المحوّل ALPHA الموجود على المسافة 512 بايتة من العنوان القاعدي . ولنشحن في

المرصف 3 مضمون العنوان ALPHA المؤشر بواسطة المرصف 5 . بإمكاننا كتابة التعليقات التالية بلغة المؤول :

- بتحليل القاعدة بشكل واضح : $L(3,512)$.
- أو $L(5)$, ALPHA 3 , القاعدة هي ضمناً مرتبطة بـ ALPHA ومعدلة بواسطه المؤول حسب التوجيه USING . يوجد عدة إمكانيات لكتابة منطقة العوامل ، وهذا ما سنقوم بشرحه الآن .

2.9 . كتابة العوامل

في الإعتبارات التالية D, X, B, R, M و L تُمثل على التوالي الإزاحة ، رقم مرصف المؤشر ، رقم مرصف القاعدة ، رقم المرصف العام ، قناع (موجود في التعليمات) والطول . الدلائل 1 ، 2 و 3 هي مرتبطة بمختلف المتغيرات . جميع هذه الرموز يجب أن تكون عبارة عن تعابير مطلقة . S ستمثل تعبيراً متحولاً يمكن أن يتحول عملياً إلى رمز واحد . وتحديد أكثر للمرصف القاعدي ، فإن عوامل (متغيرات) التعليقات يمكن أن تُكتب بلغة المؤول ، حسب النسق ، على الشكل التالي :

النسق	المعاملات
RR	R_1, R_2
RX	$R_1, D_2(X_2, B_2)$
RS	$\left\{ \begin{array}{l} R_1, R_3, D_2(B_2) \\ R_1, M_3, D_2(B_2) \end{array} \right.$
SI	$D_1(B_1), I_2$
SS	$\left\{ \begin{array}{l} D_1(L_1, B_1), D_2(B_2) \\ D_1(L_1, B_1), D_2(L_2, B_2) \end{array} \right.$
S	$D_2(B_2)$

جدول 1.9

العنوان المحوّل هو دائماً العنوان المحسوب في لحظة تنفيذ الجمع $D+X+B$ للتعليقات ذات النسق RX أو $D+B$ للتعليقات RS ، SI أو SS . العوامل $D(X, B)$ ، $D(B)$ ، و $D(L, B)$ يمكن أن تُستبدل بواسطة العوامل حيث رقم المرصف القاعدي والإزاحة سيتم حسابهما بواسطة المؤول . وستكتب إذاً على الشكل التالي : S ، $S(L)$ ، أو $S(X)$.

الجدول التالي يعرض لنا مختلف إمكانيات كتابة هذه المعاملات حسب نسق

التعليمة . سنشير هنا إلى أنه بداخل الأهلة ، وفي الشكليات مع قاعدة ضمنية أو جلية ، لا يمكن أن نجد سوى التعابير المطلقة حيث المعنى الأساسي ، الدليل أو الطول يتعلق بنسق التعليمة وبالطبيعة مطلق أو تحول للتعبير المذكور على يسار الأهلة .

أمثلة :

ABS و TRANS هما تعبيران مطلقان وعحولان . ABS1 (ABS2) في التعليمة RS يمكن أن تفهم وكأنها D(B) . ABS1 (TRANS) هي مغلوطة معها يكن النسق ، TRANS (ABS1) يجب أن تفهم كأنها S(X) في التعليمة RX وكأنها S(L) في التعليمة SS .

نسق التعليمة	الكتابة بتعابير مطلقة قاعدة جلية	الكتابة بتعابير محولة قاعدة ضمنية
RS et SI	D(B)	S
SS	D(L,B) D(B) (1) D(B)	S(L)
RX	D(X,B)	S(X) S

جدول 2.9

حالات خاصة

X أو B يعادل صفراً .

D(0) يمكن أن يكتب D

D (0,B) يمكن أن يكتب D(B)

D (X,0) يمكن أن يكتب D(X) أو D(X) . (أمثلة أنظر صفحة 82) .

3.9 . قواعد الاصطفاة أو التراصف

مع أن أولية العنونة تسمح بعنونة البايئة ، فإن عناوين متأثرات التعليمة يجب أن تخضع لبعض قواعد التوافق . قواعد كهذه هي موجودة على جميع المكتات .

تستعمل التعليقات متأثرين قد يكونان عبارة عن مرصف وعنوان من الذاكرة أو عنوانين من الذاكرة . نحدد القواعد حسب المعطيات التي تُعالجها التعليقات .

بالنسبة للتعليقات التي تُعالج كلمات - مزدوجة ، كلمات أو نصف - كلمات ، فإن

(1) الطول هو ضمني ، المؤول يختار الخاصية - طول . الطول المؤول هو دوماً الطول القملي ناقص واحد .

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
00000				1	PRINT DATA	
00000	95C D00C	0000C		2	* SEQUENCE	
00000	095D	00006		3	START	
00006	4770 C00E	00C14		4	END	
00000	6700	00000		5	BASE 12 EXPLICIT	
00000	0133867	00000		6	BASE 12 EXPLICIT	
00010	894BC0E7	00000		7	BASE 12 EXPLICIT	
00014	5035 C004	00004		8	BASE 12 EXPLICIT	
0001C	5030 C00A	00010		9	BASE 12 EXPLICIT	
00020	5000 C008	00020		10	BASE 12 EXPLICIT	
00024	5000 C008	00024		11	BASE 12 EXPLICIT	
00028	5000 C008	00028		12	BASE 12 EXPLICIT	
00032	5030 C00E	00032		13	BASE 12 EXPLICIT	
00036	5030 C00E	00036		14	BASE 12 EXPLICIT	
00040	5030 C00E	00040		15	BASE 12 EXPLICIT	
00044	5030 C00E	00044		16	BASE 12 EXPLICIT	
00048	5030 C00E	00048		17	BASE 12 EXPLICIT	
00052	5030 C00E	00052		18	BASE 12 EXPLICIT	
00056	5030 C00E	00056		19	BASE 12 EXPLICIT	
00060	5030 C00E	00060		20	BASE 12 EXPLICIT	
00064	5030 C00E	00064		21	BASE 12 EXPLICIT	
00068	5030 C00E	00068		22	BASE 12 EXPLICIT	
00072	5030 C00E	00072		23	BASE 12 EXPLICIT	
00076	5030 C00E	00076		24	BASE 12 EXPLICIT	
00080	5030 C00E	00080		25	BASE 12 EXPLICIT	
00084	5030 C00E	00084		26	BASE 12 EXPLICIT	
00088	5030 C00E	00088		27	BASE 12 EXPLICIT	
00092	5030 C00E	00092		28	BASE 12 EXPLICIT	
00096	5030 C00E	00096		29	BASE 12 EXPLICIT	
00100	5030 C00E	00100		30	BASE 12 EXPLICIT	
00104	5030 C00E	00104		31	BASE 12 EXPLICIT	
00108	5030 C00E	00108		32	BASE 12 EXPLICIT	
00112	5030 C00E	00112		33	BASE 12 EXPLICIT	
00116	5030 C00E	00116		34	BASE 12 EXPLICIT	
00120	5030 C00E	00120		35	BASE 12 EXPLICIT	
00124	5030 C00E	00124		36	BASE 12 EXPLICIT	
00128	5030 C00E	00128		37	BASE 12 EXPLICIT	
00132	5030 C00E	00132		38	BASE 12 EXPLICIT	
00136	5030 C00E	00136		39	BASE 12 EXPLICIT	
00140	5030 C00E	00140		40	BASE 12 EXPLICIT	
00144	5030 C00E	00144		41	BASE 12 EXPLICIT	
00148	5030 C00E	00148		42	BASE 12 EXPLICIT	
00152	5030 C00E	00152		43	BASE 12 EXPLICIT	
00156	5030 C00E	00156		44	BASE 12 EXPLICIT	
00160	5030 C00E	00160		45	BASE 12 EXPLICIT	
00164	5030 C00E	00164		46	BASE 12 EXPLICIT	
00168	5030 C00E	00168		47	BASE 12 EXPLICIT	
00172	5030 C00E	00172		48	BASE 12 EXPLICIT	
00176	5030 C00E	00176		49	BASE 12 EXPLICIT	
00180	5030 C00E	00180		50	BASE 12 EXPLICIT	
00184	5030 C00E	00184		51	BASE 12 EXPLICIT	
00188	5030 C00E	00188		52	BASE 12 EXPLICIT	
00192	5030 C00E	00192		53	BASE 12 EXPLICIT	
00196	5030 C00E	00196		54	BASE 12 EXPLICIT	
00200	5030 C00E	00200		55	BASE 12 EXPLICIT	
00204	5030 C00E	00204		56	BASE 12 EXPLICIT	
00208	5030 C00E	00208		57	BASE 12 EXPLICIT	
00212	5030 C00E	00212		58	BASE 12 EXPLICIT	
00216	5030 C00E	00216		59	BASE 12 EXPLICIT	
00220	5030 C00E	00220		60	BASE 12 EXPLICIT	
00224	5030 C00E	00224		61	BASE 12 EXPLICIT	
00228	5030 C00E	00228		62	BASE 12 EXPLICIT	
00232	5030 C00E	00232		63	BASE 12 EXPLICIT	
00236	5030 C00E	00236		64	BASE 12 EXPLICIT	
00240	5030 C00E	00240		65	BASE 12 EXPLICIT	
00244	5030 C00E	00244		66	BASE 12 EXPLICIT	
00248	5030 C00E	00248		67	BASE 12 EXPLICIT	
00252	5030 C00E	00252		68	BASE 12 EXPLICIT	
00256	5030 C00E	00256		69	BASE 12 EXPLICIT	
00260	5030 C00E	00260		70	BASE 12 EXPLICIT	
00264	5030 C00E	00264		71	BASE 12 EXPLICIT	
00268	5030 C00E	00268		72	BASE 12 EXPLICIT	
00272	5030 C00E	00272		73	BASE 12 EXPLICIT	
00276	5030 C00E	00276		74	BASE 12 EXPLICIT	
00280	5030 C00E	00280		75	BASE 12 EXPLICIT	
00284	5030 C00E	00284		76	BASE 12 EXPLICIT	
00288	5030 C00E	00288		77	BASE 12 EXPLICIT	
00292	5030 C00E	00292		78	BASE 12 EXPLICIT	
00296	5030 C00E	00296		79	BASE 12 EXPLICIT	
00300	5030 C00E	00300		80	BASE 12 EXPLICIT	
00304	5030 C00E	00304		81	BASE 12 EXPLICIT	
00308	5030 C00E	00308		82	BASE 12 EXPLICIT	
00312	5030 C00E	00312		83	BASE 12 EXPLICIT	
00316	5030 C00E	00316		84	BASE 12 EXPLICIT	
00320	5030 C00E	00320		85	BASE 12 EXPLICIT	
00324	5030 C00E	00324		86	BASE 12 EXPLICIT	
00328	5030 C00E	00328		87	BASE 12 EXPLICIT	
00332	5030 C00E	00332		88	BASE 12 EXPLICIT	
00336	5030 C00E	00336		89	BASE 12 EXPLICIT	
00340	5030 C00E	00340		90	BASE 12 EXPLICIT	
00344	5030 C00E	00344		91	BASE 12 EXPLICIT	
00348	5030 C00E	00348		92	BASE 12 EXPLICIT	
00352	5030 C00E	00352		93	BASE 12 EXPLICIT	
00356	5030 C00E	00356		94	BASE 12 EXPLICIT	
00360	5030 C00E	00360		95	BASE 12 EXPLICIT	
00364	5030 C00E	00364		96	BASE 12 EXPLICIT	
00368	5030 C00E	00368		97	BASE 12 EXPLICIT	
00372	5030 C00E	00372		98	BASE 12 EXPLICIT	
00376	5030 C00E	00376		99	BASE 12 EXPLICIT	
00380	5030 C00E	00380		100	BASE 12 EXPLICIT	
00384	5030 C00E	00384		101	BASE 12 EXPLICIT	
00388	5030 C00E	00388		102	BASE 12 EXPLICIT	
00392	5030 C00E	00392		103	BASE 12 EXPLICIT	
00396	5030 C00E	00396		104	BASE 12 EXPLICIT	
00400	5030 C00E	00400		105	BASE 12 EXPLICIT	
00404	5030 C00E	00404		106	BASE 12 EXPLICIT	
00408	5030 C00E	00408		107	BASE 12 EXPLICIT	
00412	5030 C00E	00412		108	BASE 12 EXPLICIT	
00416	5030 C00E	00416		109	BASE 12 EXPLICIT	
00420	5030 C00E	00420		110	BASE 12 EXPLICIT	
00424	5030 C00E	00424		111	BASE 12 EXPLICIT	
00428	5030 C00E	00428		112	BASE 12 EXPLICIT	
00432	5030 C00E	00432		113	BASE 12 EXPLICIT	
00436	5030 C00E	00436		114	BASE 12 EXPLICIT	
00440	5030 C00E	00440		115	BASE 12 EXPLICIT	
00444	5030 C00E	00444		116	BASE 12 EXPLICIT	
00448	5030 C00E	00448		117	BASE 12 EXPLICIT	
00452	5030 C00E	00452		118	BASE 12 EXPLICIT	
00456	5030 C00E	00456		119	BASE 12 EXPLICIT	
00460	5030 C00E	00460		120	BASE 12 EXPLICIT	
00464	5030 C00E	00464		121	BASE 12 EXPLICIT	
00468	5030 C00E	00468		122	BASE 12 EXPLICIT	
00472	5030 C00E	00472		123	BASE 12 EXPLICIT	
00476	5030 C00E	00476		124	BASE 12 EXPLICIT	
00480	5030 C00E	00480		125	BASE 12 EXPLICIT	
00484	5030 C00E	00484		126	BASE 12 EXPLICIT	
00488	5030 C00E	00488		127	BASE 12 EXPLICIT	
00492	5030 C00E	00492		128	BASE 12 EXPLICIT	
00496	5030 C00E	00496		129	BASE 12 EXPLICIT	
00500	5030 C00E	00500		130	BASE 12 EXPLICIT	
00504	5030 C00E	00504		131	BASE 12 EXPLICIT	
00508	5030 C00E	00508		132	BASE 12 EXPLICIT	
00512	5030 C00E	00512		133	BASE 12 EXPLICIT	
00516	5030 C00E	00516		134	BASE 12 EXPLICIT	
00520	5030 C00E	00520		135	BASE 12 EXPLICIT	
00524	5030 C00E	00524		136	BASE 12 EXPLICIT	
00528	5030 C00E	00528		137	BASE 12 EXPLICIT	
00532	5030 C00E	00532		138	BASE 12 EXPLICIT	
00536	5030 C00E	00536		139	BASE 12 EXPLICIT	
00540	5030 C00E	00540		140	BASE 12 EXPLICIT	
00544	5030 C00E	00544		141	BASE 12 EXPLICIT	
00548	5030 C00E	00548		142	BASE 12 EXPLICIT	
00552	5030 C00E	00552		143	BASE 12 EXPLICIT	
00556	5030 C00E	00556		144	BASE 12 EXPLICIT	
00560	5030 C00E	00560		145	BASE 12 EXPLICIT	
00564	5030 C00E	00564		146	BASE 12 EXPLICIT	
00568	5030 C00E	00568		147	BASE 12 EXPLICIT	
00572	5030 C00E	00572		148	BASE 12 EXPLICIT	
00576	5030 C00E	00576		149	BASE 12 EXPLICIT	
00580	5030 C00E	00580		150	BASE 12 EXPLICIT	
00584	5030 C00E	00584		151	BASE 12 EXPLICIT	
00588	5030 C00E	00588		152	BASE 12 EXPLICIT	
00592	5030 C00E	00592		153	BASE 12 EXPLICIT	
00596	5030 C00E	00596		154	BASE 12 EXPLICIT	
00600	5030 C00E	00600		155	BASE 12 EXPLICIT	
00604	5030 C00E	00604		156	BASE 12 EXPLICIT	
00608	5030 C00E	00608		157	BASE 12 EXPLICIT	
00612	5030 C00E	00612		158	BASE 12 EXPLICIT	
00616	5030 C00E	00616		159	BASE 12 EXPLICIT	
00620	5030 C00E	00620		160	BASE 12 EXPLICIT	
00624	5030 C00E	00624		161	BASE 12 EXPLICIT	
00628	5030 C00E	00628		162	BASE 12 EXPLICIT	
00632	5030 C00E	00632		163	BASE 12 EXPLICIT	
00636	5030 C00E	00636		164	BASE 12 EXPLICIT	
00640	5030 C00E	00640		165	BASE 12 EXPLICIT	
00644	5030 C00E	00644		166	BASE 12 EXPLICIT	
00648	5030 C00E	00648		167	BASE 12 EXPLICIT	
00652	5030 C00E	00652		168	BASE 12 EXPLICIT	
00656	5030 C00E	00656		169	BASE 12 EXPLICIT	
00660	5030 C00E	00660		170	BASE 12 EXPLICIT	
00664	5030 C00E	00664		171	BASE 12 EXPLICIT	
00668	5030 C00E	00668		172	BASE 12 EXPLICIT	
00672	5030 C00E	00672		173	BASE 12 EXPLICIT	
00676	5030 C00E	00676		174	BASE 12 EXPLICIT	
00680	5030 C00E	00680		175	BASE 12 EXPLICIT	
00684	5030 C00E	00684		176	BASE 12 EXPLICIT	
00688	5030 C00E	00688		177	BASE 12 EXPLICIT	
00692	5030 C00E	00692		178	BASE 12 EXPLICIT	
00696	5030 C00E	00696		179	BASE 12 EXPLICIT	
00700	5030 C00E	00700		180	BASE 12 EXPLICIT	
00704	5030 C00E	00704		181	BASE 12 EXPLICIT	
00708	5030 C00E	00708		182	BASE 12 EXPLICIT	
00712	5030 C00E	00712		183	BASE 12 EXPLICIT	
00716	5030 C00E	00716		184	BASE 12 EXPLICIT	
00720	5030 C00E	00720		185	BASE 12 EXPLICIT	
00724	5030 C00E	00724		186	BASE 12 EXPLICIT	
00728	5030 C00E	00728		187	BASE 12 EXPLICIT	
00732	5030 C00E	00732		188	BASE 12 EXPLICIT	
00736	5030 C00E	00736		189		

عناوين التأثيرات يجب أن تُصَفَّ حسب الحدود المناسبة . أما تلك التي تعالج السمات فلا يوجد أية مشكلة بالنسبة لها . إنَّ عدم المحافظة على هذه القواعد يؤدي إلى حدوث مشكلة في المؤول (أنظر المثل السطر 33) ، فهو يؤدي عند التنفيذ إلى انقطاع من نوع «Specification» (تميز) . التعليقات يجب أيضاً أن تُصَفَّ في حدود نصف كلمات .

تمارين

تمرين 1.9 - للتعليقات أدناه :

- 1- إفحص إذا كانت العناصر التي تؤلف التأثيرات هي مُطلقة أو محوَّلة .
- 2- باعتبار النسق المرتبطة بكل تعليمة نستخلص ، فقط حسب المعايير النحوية ، إذا كانت التعليقات صحيحة .
- 3- قم بإجراء تأويل التعليقات الصحيحة .

		مرصف القاعدة = 12	
		النسق	
ADBASE	CSECT	*,12	
	USING	B,D	RX
	L	3,D(3)	RX
	LR	A,D	RR
	ST	D,X'4'(3,C)	RX
	L	A,B'1011'(3)	RX
	L	D,E(B)	RX
	L	A,E(B)	RX
	MVC	A(B,C),D	SS
	MVC	E(L'D),D	SS
	L	2,D+L'D	RX
	EQU	0	
	EQU	1	
	EQU	10	
	DS	5F	
	DS	12F	
	INIT		
A			
B			
C			
D			
E			

10 . التعليمات بلغة المؤول عموميات

سنقوم بدراسة التعليمات - الآلية حسب نوع التمثيل الداخلي الذي تُعالجه هذه التعليمات . من البديهي أن تكون التعليمات الحسائية العشرية ، مثلاً ، بدون معنى إلا عندما نُقدِّم لها معطيات مكوَّنة عشرياً . مثلاً ، من الواضح أن المرافف المبلوغة بالتعليمات المتحركة هي مرافف متحركة .
سنبدأ بالتعليمات التي تعمل على المرافف العامة ، ولكن في البداية يجب عرض الترميز المعتمد .

1.10 . الترميز

سيتم تحديد التعليمات - الآلية حسب النسق التالي :

المعنى	كود العمليات ساحس عشري	النسق	العوامل	كود - العملية الرمزي
LOAD (S ₂) + R ₁	COP=58	RX	R ₁ , D ₂ (X ₂ , B ₂)	L

تشير العوامل إلى العناوين مع قاعدة محدَّدة بشكل جي . أمَّا الشروحات فتذكر هذا العنوان بشكل رمزي . فإذا S₂ ستمعي العنوان المحسوب بإضافة مضمون المرافف القاعدية والمؤشر إلى الإزاحة ، في المجموع فإن S₂=D₂+B₂+X₂ بالنسبة للتعليمات RX و S₂=D₂+B₂ للباقية .

سنجد في الحيز مُعاملات أو في الشروحات الرموز التالية :

- R₁, R₂ هي عبارة عن أرقام المرافف التي يمكن أن تُستبدل بالتعابير المطلقة .
- D قيمة الإزاحة بالنسبة إلى العنوان القاعدي .
- X رقم المرافف المؤشر المستعمل .
- B رقم المرافف القاعدي .
- M قناع من أربع بتات موجود في التعليم .

I قيمة فورية موجودة في التعليمة .
CO عدّاد البرنامج (Program counter) .
S عنوان رمزي ، تعبير قابل للتحويل :

$$S = D_2 + X_2 + B_2$$

$$S = D_2 + B_2 \text{ أو}$$

(S) مضمون العنوان S .

→ رمز للتخصيص ، أي نسخ منطقة في أخرى دون تهديم المنطقة الأصلية . مثلاً :
(S) → R₁ يعني نسخ مضمون المرصف R₁ في المنطقة من الذاكرة بالعنوان S . لن
نستعمل أبداً الترميز (R) للإشارة إلى مضمون المرصف R لأنه لن يوجد أي
إبهام ، في حالة المرصف يتعلّق ذلك دائماً بالمضمون بينما يجب التمييز بين الإسم
S للذاكرة ومضمونها .

((S)) من الممكن استعمال هذا التعبير للإشارة إلى أن مضمون العنوان S هو نفسه المعتمد
كعنوان نأخذ منه المضمون .

CC يعني كود - الشرط .
الدلائل (indices) الدلائل 1 ، 2 ، 3 تُرجع إلى الحقول المرتبطة بالتعليمة الآلية (فقرة
1.5) .

(24-31) R₁ تعني البتات 24 إلى 31 من المرصف رقم R₁ .
R₁ ، R₁₊₁ تعني المرصف المزدوج المؤلف من المرافف ذات الرقم R₁ و R₁₊₁ . R₁ يكون
رقماً مزدوجاً .

العناوين (adresses) تشير إلى أن العناوين تعني البايته من اليسار لمنطقة ما ، وإن البتات
من الكلمة ، من مرصف ... هي مرقّمة من اليسار إلى اليمين إنطلاقاً
من 0 .

(370) تشير إلى أن التعليمة غير موجودة إلا على المكتبة 370 :

2.10 . كود العمليات الحرفية التذكيرية

كتابة كود - العمليات الرمزية يخضع إلى قواعد من المفيد الإشارة لها هنا . إن كود
العملية يترجم الفعل المطلوب إجراؤه . السمة الأولى (أحياناً السمتان الأوليان) هي
بداية الفعل الذي يُعبّر عن العمل .

مثلاً :

A	Add	جمع
L	LOAD	شحن

ST	STore	خزن
MVC	MoVe	نقل

الأحرف التالية هي معدلات (1) أو أنها تُحْمِز نوع المعطيات المُعالِجَة (2) أو أيضاً النسق RR أو SI للتعليقات (3) .

أمثلة :

(1)	AL	جمع منطقي Add Logical
(2)	CVB	تحويل إلى ثنائي ConVert Character
(2)	AE	جمع معطيات من نوع بفاصلة متحركة قصير Add données de type E (flottant court)
(2)	MVC	نقل السمات MoVe Characters
(2)	AD	جمع معطيات من نوع D
(3)	LR	شحن بنسق RR
(3)	LPR	شحن إيجابي بنسق RR
(3)	MVI	شحن مباشر بنسق SI

11 . الحساب بفاصلة ثابتة والحركات

- 1.11 . تعليمات الشحن والتخزين في المراسف العامة
هذه هي التعليمات التي تنسخ المتأثر في أحد المراسف :
« عنوان المتأثر , رقم المراسف LOAD »
وتنسخ مضمون المراسف في الذاكرة على عنوان معين :
« عنوان , رقم المراسف STORE »

هذه العمليات لا تؤثر على المتأثر الأساسي . بعض التعليمات تؤدي إلى تركيز كود الشرط CC , لموقعين ثنائيين يتميان إلى PSW (فصل 4) , تبعاً لإشارة المتأثر المنقول حسب الاتفاق التالي :

- بعد العملية فإن CC سيُرَكِّز على (1) . :
- 0 إذا كانت النتيجة صفراً .
- 1 إذا كانت النتيجة سلبية .
- 2 إذا كانت النتيجة إيجابية .
- 3 إذا كان هناك زيادة عن السعة (overflow) .

الزيادة عن السعة تؤدي عادة إلى إنقطاع في تنفيذ البرنامج . أي أنه سيحدث خطأ يُعالجه نظام التشغيل . يوجد برنامج ، يُدعى برنامج إنقطاع «fixed point overflow» ، يعطي العلاج للمستعمل ويوقف العمل في تنفيذ البرنامج بنهاية غير طبيعية . بإمكان المبرمج أن يقوم بتفنيح عملية الإنقطاع هذه في بعض الحالات بتركيز البتات المناسبة لقناع البرنامج في PSW .
وسندرس هذا الأمر لاحقاً (الفصل 19) .

(1) هذا الاتفاق هو صالح فقط للتعليمتين LOAD وSTORE وبعض التعليمات الأخرى . وسنرى كيف يتم تركيز CC لكل مجموعة تعليمات .

المُتأثر 1 هو دائماً مرصّف ، والمُتأثر الثاني يُمكن أن يكون مرصفاً ، نصف كلمة أو كلمة - ذاكرة .

من المهم أن نشير إلى أن المُتأثرات الموجودة على العناوين المشار إليها بواسطة S يجب أن تُحصر في حدود كلمات أو نصف - كلمات حسب التعليقات .

LR	R_1, R_2	RR	COP=18	LOAD
				$R_2 \rightarrow R_1$
L	$R_1, D_2(X_2, B_2)$	RX	COP=5B	LOAD
				$(S_2) \rightarrow R_1$
		CC	لا يتغيّر	

LH	$R_1, D_2(X_2, B_2)$	RX	COP=48	LOAD HALFWORD
				$(S_2) \rightarrow R_1$

يُعتبر المُتأثر الثاني كعدد صحيح بإشارة وبطول 16 بتة . يُوسّع إلى 32 بتة قبل التحويل. CC لا يتأثر .

LCR	R_1, R_2	RR	COP=13	LOAD COMPLEMENT
				$R_2 \rightarrow R_1$

يُجزّن عكس (مكمّل إلى 2) R_2 في R_1 . overflow إذا أكملنا العدد السليبي الأقصى . يوضع CC حسب الإشارة النهائية لـ R_1 .

LPR	R_1, R_2	RR	COP=10	LOAD POSITIVE
				$R_2 \rightarrow R_1$

القيمة المطلقة لـ $R_2 \rightarrow R_1$ سيحدث زيادة عن السعة (overflow) إذا أكملنا العدد السليبي الأقصى . يُركّز CC على 0 ، 2 أو 3 حسب النتيجة .

LNR	R_1, R_2	RR	COP=11	LOAD NEGATIVE
-----	------------	----	--------	---------------

للكمّل إلى 2 للقيمة المطلقة لـ R_2 يُجزّن في R_1 . لن يحدث overflow . CC يُركّز على 0 أو 1 .

LTR	R_1, R_2	RR	COP=12	LOAD AND TEST
				$R_2 \rightarrow R_1$

تعليمة شبيهة بـ LR باستثناء كون الإشارة النهائية لـ R_1 تُركّز CC . R_1 يُمكن أن يكون معادلاً لـ R_2 .

LM	$R_1, R_3, D_2(B_2)$	RS	COP=88	LOAD MULTIPLE
----	----------------------	----	--------	---------------

الواقع التالي للذاكرة ، انطلاقاً من العنوان S2 ستشتمل في المرافف العامة R_1, R_{1+1}, \dots, R_n في هذه التعليمة يُفترض بأن يتبع المرصّف 0 المرصّف 15 . هكذا :

LM 15, 1, ALPHA ستشتمل الكلمة ذات العنوان ALPHA في المرصّف 15 ، وذلك ذات العنوان ALPHA+4 في المرصّف 0 وهكذا دواليك . تُستخدم هذه التعليمة بشكلٍ خاص لتزويد إطار البرنامج CC لا يتأثر .

LA $R_1, D_2(X_2, B_2)$ RX COP=41 LOAD ADDRESS

$S_2 \rightarrow R_1(8-31) \quad 0 \rightarrow R_1(0-7)$

تُخزَّن القيمة ذات العنوان S_2 في البتات من 8 إلى 31 من المرصف R_1 .
يتم تصفير البتات من 0 إلى 7. وتطبق هنا قواعد حساب العنوان،
أي أن القيمة $D_2 + X_2 + B_2$ تُخزَّن (عنوان فعلي). من الممكن أن
تأخذ نفس المرصف لـ R_1, X_2 أو B_2 . المرصف 0 لا يؤخذ أبداً وكانه
قاعدة أو مرصف تأثير.
الاستعمال: أنظر التمارين
- شحن عنوان في مرصف،
- شحن عدد غير سلمي أصغر أو يعادل 4095 (القيمة القصوى للإزاحة)
في مرصف،

- زيادة مضمون مرصف بقيمة أصغر أو تساوي 4095.

IC $R_1, D_2(X_2, B_2)$ RX COP=43 INSERT CHARACTER

$(S_2) \rightarrow R_1(24-31)$

لا يتغيَّر $R_1(0-23)$

يتم تخزين بايت واحدة بعنوان S_2 في R_1 . CC لا يتأثر.

ICM $R_1, M_3, D_2(B_2)$ RS COP=BF INSERT CHARACTERS UNDER MASK (370)

تربط البتات الأربع من القناع M_3 بالبتات الأربع للمرصف R_1 .
البايتات من R_1 المرتبطة بالبتات «1» من القناع يتم شحنها مع البايتات
للتبائية من S_2 . طول التأثير الثاني يعادل عدد «1» في القناع.
يُركِّز كود الشرط:

CC = 0 : جميع البتات الداخلة هي مصفَّرة أو القناع مصفَّر،

CC = 1 : البتة ذات الوزن الأكبر في S_2 هي «1»،

CC = 2 : البتات ذات الوزن الأكبر في S_2 هي «0» ولكن جميع البتات
الداخلة ليست صفراً.

وفي الختام فإن CC يُركِّز حسب إشارة S_2 .

ST $R_1, D_2(X_2, B_2)$ RX COP=50 STORE
 $R_1 \rightarrow (S_2)$

CC والمرصف R_1 يتقيان بدون تعديل.

STH $R_1, D_2(X_2, B_2)$ RX COP=40 STORE HALFWORD
 $R_1(16-31) \rightarrow (S_2)$

التأثير الثاني هو بطول 2 بايتة. CC يبقى بدون تعديل.

STM $R_1, R_3, D_2(B_2)$ RS COP=90 STORE MULTIPLE

للمراصف العادة من R_1 إلى R_3 يتم تخزينها في مواقع متتالية من الذاكرة
بدءاً من العنوان S_2 . الرقم 0 للمرصف 0 مفترض أنه يتبع الرقم 15
بشكل يؤدي معه تنفيذ التعليمة ALPHA 15, 1, ST إلى تخزين

المرافق 15 ، 0 ، 1 ، بالمتاوين ALPHA+4 ، ALPHA ، ...
تستخدم التعليمة بشكل خاص لحفظ إطار البرنامج . CC يبقى بدون
تغيير .

STC $R_1, D_2(X_2, B_2)$ RX COP=42 STORE CHARACTER
 $R_1(24-31) \rightarrow (S_2)$

R_1 و CC يقيان بدون تعديل .

STCM $R_1, M_3, D_2(B_2)$ RS COP=5E STORE CHARACTERS UNDER MASK
(370)

البتات الأربع من القناع M_3 ترتبط بالأربع بايتات من المرصف R_1 .

لما بايتات R_1 ، والمختارة بوجود «1» في القناع ، فيتم تخزينها بشكل
متناس على العنوان S_2 . كود الشرط CC لا يتغير .

2.11 . التعليقات الحسابية بفاصلة ثابتة

هي التعليقات التي تعمل على معطيات ممثلة بفاصلة ثابتة . تكوّن القيم السلبية
بواسطة المكمل إلى 2 . كما تقوم بالعمليات الأربع الأساسية بين مرصف ومرصف أو
بين مرصف وذاكرة . الضرب والجمع يستعملان مرافق مزدوجة (فقرة 1.10) . هذه
التعليقات تؤدي إلى تعديل CC حسب إشارة النتيجة ، وحسب الإتفاق الجاري كما في
1.11 .

$0 = CC$ إذا كانت النتيجة صفراً .

$1 = CC$ إذا كانت النتيجة سلبية .

$2 = CC$ إذا كانت النتيجة إيجابية .

$3 = CC$ إذا كان هناك overflow .

يمكن قطع التعليمة في حالة حدوث حادثة غير طبيعية ، كما يلي :

- عنوان من خارج المنطقة المخصصة .

- جهة متأثر غير صحيحة ، مرصف مزدوج معني بشكل ممي .

- فيض عن السعة overflow .

AR R_1, R_2 RR COP=1A ADD
 $R_1 + R_2 \rightarrow R_1$

A $R_1, D_2(X_2, B_2)$ RX COP=5A ADD
 $R_1 + (S_2) \rightarrow R_1$

لا يتغير المتأثر الثاني . يتم تركيز كود الشرط CC ، إحتيال

حصول overflow .

AH $R_1, D_2(X_2, B_2)$ RX COP=4A ADD HALFWORD
 $(S_2) + R_1 \rightarrow R_1$

المؤثر (S₂) هو على نصف كلمة . يُوسَّع إلى كلمة قبل العملية . يتم تركيز CC . احتمال حصول Overflow .

SR R_1, R_2 RR COP=1B SUBTRACT
 $R_1 - R_2 \rightarrow R_1$

S $R_1, D_2(X_2, B_2)$ RX COP=5B SUBTRACT
 $R_1 - (S_2) \rightarrow R_1$

المؤثر الثاني لا يتعمل . يتم تركيز CC .

SH $R_1, D_2(X_2, B_2)$ RX COP=4B SUBTRACT HALFWORD
 $R_1 - (S_2) \rightarrow R_1$

المؤثر S₂ هو على نصف كلمة ، يُوسَّع إلى 32 بتة قبل العملية . يتم تركيز CC .

MR R_1, R_2 RR COP=1C MULTIPLY
 $R_{1+1} \times R_2 \rightarrow R_1, R_{1+1}$

M $R_1, D_2(X_2, B_2)$ RX COP=5C MULTIPLY
 $R_{1+1} \times (S_2) \rightarrow R_1, R_{1+1}$

المرفص R₁ المذكور في التعليمة يجب أن يكون مرفصاً مزدوجاً . المؤثر الأول يجب أن يكون موجوداً في R₁₊₁ وعصوياً لجهة الشال . النتيجة ستوضع في R₁ ، R₁₊₁ . لا احتمال لحوث overflow ، لا يتم تركيز CC .

MH $R_1, D_2(X_2, B_2)$ RX COP=4C MULTIPLY HALFWORD
 $R_{1+1} \times (S_2) \rightarrow R_1, R_{1+1}$

المرفص R₁ يجب أن يكون مرفصاً مزدوجاً ، S₂ يتألف من 16 بتة ويُعتبر كعدد صحيح بإشارة يُوسَّع إلى 32 بتة قبل العملية . لا يحدث overflow ولا يتم تركيز CC .

DR R_1, R_2 RR COP=1D DIVIDE
 $R_1, R_{1+1} : R_2 \rightarrow \begin{matrix} R_1 & \text{بقي} \\ R_{1+1} & \text{قيمة القسمة} \end{matrix}$

R₁ هو مرفص مزدوج . يتمتع الباقي بنفس إشارة المقسوم . عندما لا تقع 32 بتة نتيجة القسمة يحدث overflow . لا يتم تركيز CC .

D $R_1, D_2(X_2, B_2)$ RX COP=5D DIVIDE
 $R_1, R_{1+1} : (S_2) \rightarrow \begin{matrix} R_1 & \text{reste} \\ R_{1+1} & \text{quotient} \end{matrix}$

R₁ يجب أن يكون مرفصاً مزدوجاً . للباقي نفس إشارة المقسوم . عندما لا تقع 32 بتة نتيجة القسمة يكون هناك فيض عن السعة . لا يتم تركيز CC .

ملاحظات :

دراسة هذه التعليقات تسفح لنا بملاحظة إن النتيجة محل دائماً مكان المتأثر الأول الذي يوضع منا . بينما لا يتم تعديل المتأثر الثاني . التعليقات التي تجري على نصف كلمة تفترض توسيع نصف الكلمة إلى كلمة قبل العملية .

3.11 . عمليات المقارنة بفاصلة ثابتة

تؤثر تعليمات المقارنة فقط على مضمون كود الشرط . هذه التعليقات هي خاصة حسب نوع تمثيل المعطيات المقارنة . سندرس هنا تلك المتعلقة بالفاصلة الثابتة . كما في التعليقات التي رأيناها، فإن المتأثر الأول هو دائماً موجود في مرصف معين والمتأثر الثاني في مرصف آخر أو في الذاكرة . يجري تركيز CC حسب الطريقة التالية :

- CC = 0 إذا كان المتأثر الأول = المتأثر الثاني .
- CC = 1 إذا كان المتأثر الأول أصغر من المتأثر الثاني .
- CC = 2 إذا كان المتأثر الأول أكبر من المتأثر الثاني .
- CC = 3 لا يُستعمل .

CR	R ₁ ,R ₂	RR	COP=19	COMPARE
C	R ₁ ,D ₂ (X ₂ ,B ₂)	RX	COP=59	COMPARE

المقارنة هي جبرية وتتعلق بـ 32 بتة . يتم تركيز مضمون CC .

CH	R ₁ ,D ₂ (X ₂ ,B ₂)	RX	COP=49	COMPARE HALFWORD
----	--	----	--------	------------------

يوسع المتأثر الثاني إلى 32 بتة قبل المقارنة مع إنتشار بتة الإشارة . يتم تركيز CC .

4.11 الجمع والطرح المنطقي

نعني بالجمع والطرح المنطقي، تعليقات تعدل مضمون CC بطريقة مختلفة عن الجمع والطرح العادي الذي رأيناه أعلاه . إضافة لذلك فإن overflow لا يؤدي إلى قطع البرنامج

يتم تركيز CC على الشكل التالي :

- CC = 0 إذا كانت النتيجة صفراً بدون مرّحل .
- CC = 1 إذا كانت النتيجة مختلفة عن 0 بدون مرّحل (no carry)
- CC = 2 إذا كانت النتيجة صفراً مع مرّحل .
- CC = 3 إذا كانت النتيجة مختلفة عن صفر مع مرّحل .

ALR R_1, R_2 RR COP=1E ADD LOGICAL
 $R_2 + R_1 \rightarrow R_1$

AL $R_1, D_2(X_2, B_2)$ RX COP=5E ADD LOGICAL
 $(S_2) + R_1 \rightarrow R_1$

SLR R_1, R_2 RR COP=1F SUBTRACT LOGICAL
 $R_1 - R_2 \rightarrow R_1$

SL $R_1, D_2(X_2, B_2)$ RX COP=5F SUBTRACT LOGICAL
 $R_1 - (S_2) \rightarrow R_1$

5.11 . التحريك من الذاكرة إلى الذاكرة

تتم في أغلب الأحيان بواسطة تعليمات من نوع SS . لا يوجد أي تقييد فيما يتعلق بالاصطفاف (alignment) . يمكن أن يتم تركيز الطول بشكل واضح في التعليمات : MVC ZONE 1, ZONE 2 أو ضمناً MVC ZONE 1 (L), ZONE 2 . يقوم عندها المؤول باختيار خاصية - الطول الخاصة بالتأثير الأول L'ZONE 1 . الطول المؤول هو الطول المذكور في التعليمات ناقص 1 . يمكن للمتأثرين أن يتراكبا ، ونجد هذه الميزة مستعملة في التمرين 6.11 .

MVI $D_1(B_1), I_2$ SI COP=92 MOVE
 $I_2 \rightarrow (S_1)$

يتم تخزين البلية المباشرة I_2 في S_1 .

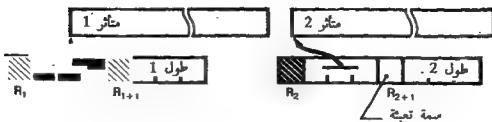
MVC $D_1(L, B_1), D_2(B_2)$ SS COP=D2 MOVE

$(S_1) \rightarrow (S_2)$ بطول L .

الحركة تتم من اليسار إلى اليمين . العملية هي غير قابلة للانقطاع عند نقل بايتين . يسمح بالتراكب وفي هذه الحالة يحدد الانتباه إلى أن الحركة تجري من اليسار إلى اليمين من أجل الحصول على النتيجة .

MVCL R_1, R_2 RR COP=0E MOVE LONG
 (370)

نسخ التأثير الثاني في التأثير الأول .
 R_1 (8-31) يحتوي على عنوان التأثير الأول ،
 $R_1 + 1$ (8-31) طول التأثير الأول ،
 R_2 (8-31) عنوان التأثير الثاني ،
 $R_2 + 1$ (8-31) سمة تسمية ،
 R_2 (8-31) طول التأثير الثاني .



الحركة تتم من اليسار إلى اليمين ، لكل بايتة على حدة . التعليمية هي قابلة للانقطاع عند نسخ بايتين . إذا كان طول المتأثر الثاني هو أصغر من طول المتأثر الأول ، يتم تكملة المتأثر الأول بسمة تعبع . يمكن تراكب المناطق بشرط أن لا يقوم النسخ بتعديل بايتة جرى تعديلها سابقاً .

يجري تركيز CC على الشكل التالي :

CC = 0 إذا كان كلا المتأثرين بنفس الطول ،

CC = 1 المتأثر الأول هو أقصر ،

CC = 2 المتأثر الأول هو أطول ،

CC = 3 إذا أدت عملية التتاليين إلى تعديل في بايتة معثلة أصلاً .

يمكن إستعمال هلم التعليمية لتصغير الذاكرة .

MVN D₁(L,B₁),D₂(B₂) SS COP=D1 MOVE NUMERIC

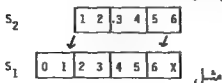
نسخ نصف - بايتات بالوزن الأضعف من (S₂) في أنصاف - بايتات الوزن الأضعف من (S₁) . تبقى أنصاف - البايتات بالوزن الأقوى دون تعديل . يسمح بالتراكب وهذا الصدد تعطي الملاحظة نفسها كما بالنسبة لـ MVC .

MVZ D₁(L,B₁),D₂(B₂) SS COP=D3 MOVE ZONES

نسخ نصف بايتات بالوزن الأقوى من (S₂) في نصف بايتات الوزن الأقوى من (S₁) . تبقى أنصاف - البايتات بالوزن الأضعف دون تعديل . يسمح بتراكب المحيزات وهذا الصدد تعطي الملاحظة نفسها كما بالنسبة لـ MVC .

MVO D₁(L₁,B₁),D₂(L₂,B₂) SS COP=F1 MOVE WITH OFFSET

نسخ من (S₂) في (S₁) مع إزاحة إلى اليسار مقدار نصف بايتة . العملية تتم من اليمين إلى اليسار ، بايتة بعد بايتة . لا يتم تغيير آخر بايتة لجهة اليمين .



تمارين

تمرين 1.11 - ضع في الصفر الثنائي أحد المرافف (أعط حلين لتعليلة واحدة دون حجز ثوابت) .

تمرين 2.11 - غير إشارة المرصف (تمثيل ثنائي) .

تمرين 3.11 - ضع جميع بتات المرصف في 1 .

تمرين 4.11 - اشحن القيمة 2048 في مرصف ، ثم القيمة 4095 (دون حجز ثابتة) بعد ذلك اشحن 4096 .

تمرين 5.11 - زد مضمون أحد المرافف مقدار 4 .

تمرين 6.11 - عيئ منطقة بطول $L \geq 256$ بايتة بنجوم (تعليماتان) .

12 التفرعات

نفهم بالتفريع كل تعديل في مضمون عداد البرنامج يؤدي إلى إنقطاع في الدوران المتتالي للعمليات .

- عودتنا دراسة اللغات المتطورة على اعتبار نوعين من الإنقطاعات في المتتالية :
- الإنقطاعات الإلزامية (GOTO في لغة فورتران) .
- الإنقطاعات المشروطة (IF) .

في لغة المؤلف ، فإن الإنقطاعات المشروطة تنتج إما عن اختيار لقيمة مأخوذة من كود الشرط ، إما عن اختبار لقيمة مأخوذة من مرصف عام . التعليمات BC و BCR تفحصان كود الشرط CC والعمليات BCT ، BCTR ، BXH ، BXLE تُفحص أو تزيد من مضمون مرصف وبعد ذلك تفحص قيمته . يمكن تنفيذ الإنقطاعات الإلزامية بواسطة BC و BCR .

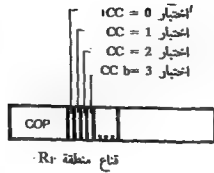
1.12 . الكود - الشرط

لقد التقيناه عند دراسة العمليات السابقة . ونذكر بأنه عبارة عن مؤشر بموقعين ثنائيين ، ينتميان إلى PSW (البتان، 34 ، 35) ويركزان بواسطة بضع عمليات حسب النتيجة الحاصلة . العمليات الحسابية ، مثلاً ، التركيز حسب إشارة النتيجة ، عمليات المقارنة حسب القيمة النسبية للمتأثرين .

الكود الشرطي CC يمكن أن يأخذ إذن أربع قيم ثنائية 00 ، 01 ، 10 ، 11 يتم مراجعتها في العمليات بواسطة 0 ، 1 ، 2 ، 3 .

2.12 . العمليات التي تفحص الكود الشرطي (CC) : BCR و BC

هذه العمليات تستعمل المنطقة R_i المكوّنة من أربع بتات ثنائية ، من نسخها الآلي ، ليس كرقم مرصف بل كقناع : كل بتة تعادل 1 وموجودة في هذه المنطقة تناسب اختبار إحدى القيم الأربع التي نحصل عليها بواسطة CC حسب الإتفاق التالي :



هكذا ، فالقناع المُعادل 1100 (ثنائياً) سيسمح باختبار الشروط $CC=0$ أو $CC=1$. الشرط المُختار فعلاً يتعلّق إذاً بالتعلّمة التي أدت إلى تركيز CC .
لقد رأينا أنّ CC تركّز حسب الطريقة التالية :

كود الشرط	0	1	2	3
تعلّيمات حسابية نتيجة	$=0$	<0	>0	ليُض من السعة
تعلّيمات مقارنة متأثر أول	$=$	$<2^*$	$>2^*$	---

القناع المُعادل لـ 1100 (أي C بالنظام السادس عشري أو 12 بالعشري) يناسب الاختبارات التالية :

- نتيجة سلبية أو صفر بعد تعلّمة حسابية .
- متأثر أول أصغر من المتأثر الثاني بعد تعلّمة مقارنة .

BCR M_1, R_2 RR COP=07 BRANCH ON CONDITION

M_1 هي القناع المذكور أعلاه .

بعد تنفيذ الشرط ، هناك تفريع إلى العنوان المخزّن في R_2 . وإلا سيتابع التنفيذ بالتوالي . مما يترجم على الشكل التالي : الشرط المتخذ $R_2 \rightarrow CO$
وإلا $CO + 2 \rightarrow CO$

BC $M_1, D_2(X_2, B_2)$ RX COP=47 BRANCH ON CONDITION

M_1 قناع .

إذا تم تنفيذ الشرط فسيحدث تفرع إلى العنوان $D_2 + X_2 + B_2$
وإلا فإن التنفيذ سيتابع بالتوالي ، مما يترجم على الشكل التالي :

في حال تنفيذ الشرط : $D_2 + X_2 + B_2 \rightarrow CO$

وإلا $CO + 4 \rightarrow CO$

$D_2 + X_2 + B_2$ عنوان التفرع ..

في لغة المؤول ، يُحدّد القناع M_1 بواسطة تعبير مطلق ، عادة رقم عشري .
BC 15, ALPHA أو BCR 15,R يناسبان القناع 1111 . يتعلّق ذلك إذا
بالتفرع المنتظم لأنه ممّا تكّن قيمة CC هناك تفرع .

BCR 0,R أو BC 0,ALPHA هي عبارة عن تعليقات دون فعل لأنه لن يتم
اختبار أي شرط . وهي تتميّز بأنها بدون فعل .

الأكواد الحرفية التذكيرية الموسّعة
وفي النهاية كي يتم تفادي تحديد القناع الخاص ولتذكر الإتفاقات المذكورة أعلاه ،
فإن المؤول يسمح باستعمال كود حرفي حسب الشرط المفحوص .

ويقوم بمهمة ترجمة الكود الحرفي إلى BC أو BCR .

هكذا :

$D_2(X_2, B_2)$ B يناسب تفرعاً غير شرطي
BC 15, $D_2(X_2, B_2)$

R_2 BR يناسب تفرعاً غير شرطي
BCR 15, R_2

$D_2(X_2, B_2)$ BNE يناسب تفرعاً معيّناً وإلا يعادل
BC 7, $D_2(X_2, B_2)$

سنجد في الملحق اللائحة الكاملة للكود الحرفي التذكيري الموسّع . سنلاحظ إن
الأكواد الحرفية تتعلّق بالتعليمة التي تقوم بتركيز الكود الشرطي . من المفيد ، لوضوح
البرنامج ، استعمال هذه الأكواد الحرفية التذكيرية . ونركّز على كون هذه الأكواد العملية
لا تتناسب سوى مع 2 كود - مكنة . ونشير ، كما ذكرنا في الفقرة 2.10 ، إلى أن الأكواد
التي تنتهي بـ R تناسب تعليقات بنسق RR أو BCR .

3.12 . . التعليقات التي تفحص القيمة المأخوذة من مرصيف (مؤشر)
أربع تعليقات BCT ، BCTR ، BXH و BXLE تسمح بتعديل مضمون

المصرف والتفريع إلى عنوان معين عندما تصبح قيمته معادلة ، أقل أو أكبر من كمية محددة .

BCTR R_1, R_2 RR COP=06 BRANCH ON COUNT
 $R_1 - 1 \rightarrow R_1$

إذا كانت $R_1 \neq 0$: $R_2 \rightarrow CO$ (تفريع إلى العنوان الموجود في R_2) .
 وإلا $CO + 2 \rightarrow CO$ (تنفيذ التعليمة التالية) .
 ملاحظة : إذا كان R_2 هو المصرف 0 فالعُد يتم بدون تفريع .

BCT $R_1, D_2(X_2, B_2)$ RX COP=46 BRANCH ON COUNT
 $R_1 - 1 \rightarrow R_1$

إذا : $R_1 \neq 0$: $S_2 \rightarrow CO$ (تفريع إلى العنوان S_2)
 وإلا : $CO + 4 \rightarrow CO$ (تنفيذ التعليمة التالية) .

BXH $R_1, R_2, D_2(B_2)$ RS COP=86 BRANCH ON INDEX HIGH

1- زيادة مضمون $R_1 : R_1 + R_2 \rightarrow R_1$
 2- عندما تصبح R_1 أكبر من المرجعية : تفريع . المرجعية هي R_{3+1}

1- R_3 هو مصرف برقم مفرد .
 R_3 هو مرجع المقارنة والزيادة .

فإذا : $R_1 + R_2 \rightarrow R_1$ بعد ذلك ، إذا كان $R_1 > R_3$
 عندئذ $S_2 \rightarrow CO$ (تفريع إلى S_2)
 وإلا $CO + 4 \rightarrow CO$ (متابعة على التوالي) .

ب- R_3 هو مصرف برقم مزدوج

نستعمل المصرف المزدوج R_3 و R_{3+1}
 R_3 هو الزيادة و R_{3+1} هو المرجعية . إذن $R_1 + R_2 \rightarrow R_1$ ثم
 إذا كان $R_1 > R_{3+1}$ عندئذ $S_2 \rightarrow CO$ (التفريع إلى S_2) .
 وإلا $CO + 4 \rightarrow CO$ (متابعة التالية) .

ملاحظة :

يجب أن لا نخلط هنا بين المصطلح إشارة مع مصرف المؤشر
 للتعليقات RX .
 المقارنة تتم جبرياً . ويتم إهمال overflow عند الجمع .

BXLE $R_1, R_2, D_2(B_2)$ RS COP=87 BRANCH ON INDEX LOW OR EQUAL

1- زيادة $R_1 : R_1 + R_2 \rightarrow R_1$

2- عندما يصبح R_1 أصغر أو يعادل المرجعية : تفريع للرجعية . المرجعية هي R_3 أو R_{3+1} .

أ- R_3 هو مرصف برقم مفرد .

R_3 هو مرجعية المقارنة والزيادة .

فإذا : $R_1 \rightarrow R_1 + R_3$ بعد ذلك ، إذا كان $R_1 \leq R_3$ عندئذ $CO \rightarrow CO$ (تفريع إلى S_2) وإلا $CO \rightarrow CO + 4$ (متابعة المتتالية)

ب- R_3 هو مرصف برقم مزدوج .

R_3 هو الزيادة ، R_{3+1} هو المرجعية .

فإذا $R_1 \rightarrow R_1 + R_3$ ثم إذا كان $R_1 \leq R_{3+1}$ عندئذ $CO \rightarrow CO$ (تفريع إلى S_2) وإلا $CO \rightarrow CO + 4$ (متابعة المتتالية) .

ملاحظة : يجب أن لا نخلط هنا بين المصطلح مؤشر مع مرصف المؤشر للعمليات RX . تتم المقارنة جبرياً . يتم إكمال overflow عند الجمع .

4.12 . تفريع مع عودة

مشكلة التفريع مع تخزين عنوان التعليمة التي تلي تعليمة التفريع تحدث عند دعوة برنامج ثانوي . هناك تعليمتان $BALR$ و BAL موجّهتان لهذا الإستعمال .

$BALR \quad R_1, R_2 \quad RR \quad COP=05 \quad \text{BRANCH AND LINK}$
 $CO \rightarrow R_1(8-31)$ (تخزين عنوان العودة)
 $CC \rightarrow R_1(0-7)$
 $R_2(8-31) \rightarrow CO$ (تفريع)

ملاحظة :

نذكر بأن قيمة عداد البرنامج CO تتغير خلال تنفيذ التعليمة .
هكذا ، فعنوان التعليمة التالية حسب $BALR$ هو المخزن في R_1 .
هناك $BALR \quad R_1, 0$ يقوم بتخزين العنوان التالي في R_1 ولكن لا تفريع . هناك إذن تنابع للمتتالية . هذا الشكل هو الأكثر استعمالاً لشعن مرصف قاعدي بالقيمة التالية لعداد البرنامج .

إذا كانت التعليمة $BALR$ موجودة على العنوان 50000 ، فإن القيمة 50002 ستخزن في R_1 .

$BAL \quad R_1, D_2(X_2, B_2) \quad RX \quad COP=45 \quad \text{BRANCH AND LINK}$
 $CO \rightarrow R_1(8-31)$ (تخزين عنوان العودة)
 $CC \rightarrow R_1(0-7)$
 $S_2 \rightarrow CO$ (تفريع إلى العنوان S_2)

كما في $BALR$ ، فعنوان التعليمة التالية سيخزن في R_1 . إذا كانت BAL موجودة على العنوان 50000 فإن مضمون R_1 هو 50004 .

EX R₁,D₂(X₂,B₂) RX COP=44 EXECUTE

هذه التعليمة تسمح بتنفيذ تعليمة واحدة موجودة خارج التابع الطبيعي للعنوان S₂. بعد ذلك ، فإن العمل يُعاود بالتوالي .

يتم تنفيذ عملية « أو » متضمنة بين البتات (R1(24-31) و (S2(8-15)) تسمح بتعديل هذا الحقل من التعليمة (رقم المرصف ، قيمة ثنائية أو طول) . إذا كان R₁ هو المرصف 0 فلا يتم تنفيذ العملية « أو » (OR) . كما لا يمكن تنفيذ عملية التحويل .

تطبيق

عندما نرغب بإجراء نقل للمعلومات MVC من منطقة لا نعرف طولها إلا في لحظة التنفيذ . هذه الحالة تحدث عند معالجة التسجيلات بطول متغير ، يكون طول الفقرة موجوداً في رأسها . من الممكن إذا تنفيذ التعليمة «MVC» . والطريقة هي التالية : شحن الطول في R1(24-31) :

BCTR	R ₁ , 0	(تنفيذ 1)
EX	R ₁ ,MOVE	
..	
MOVE	MVC

تُنفَّذ MVC مع الطول المطلوب دون أن يكون هناك تعديل للتعليمة في الذاكرة . التعليمة MVC لا تتعدّل إلا خلال مدة التنفيذ . ويمكن أن تكون موجودة في أي مكان ولكن يُفضّل أن تكون EX و MVC موجودتين في نفس الصفحة من الذاكرة كي لا تقع في خطأ محتمل في نفس الصفحة .

تمارين :

تمرين 1.12 . أكتب متتالية التعليقات التي تسمح بتكرار N مرة إحدى عمليات المعالجة .

تمرين 2.12 . لحسب مجموع عناصر جدول من الكلمات يحتوي على أعداد بفاصلة ثابتة .

تمرين 3.12 . إكسكس سلسلة من السيات CH1 في CH2 .

تمرين 4.12 . نقص مضمون المرصف 1 (تعليمة واحدة) .

تمرين 5.12 . إشحن مرصفاً معيّناً بالعنوان الجاري زائد 2 .

13 . العمليات المنطقية

1.13 . الدوال المنطقية

يسمح الكمبيوتر IBM 360/370 بعنونة البايته ، ومن غير الممكن الإشارة إلى بته معينة داخل البايته . ولكن بسبب وجود تعليمات الإزاحة (Shift) والتعليمات المنطقية سيكون بإمكاننا إختبار أو تعديل مضمون إحدى البتات من داخل الكلمة .

العمليات المنطقية الموجودة هي « و » (AND) ، الجمع « أو » (OR) و « أو المقتصرة » (EOR) . جدول العمليات المنطقية هو التالي :

A	1	0	1	0	تعليمات
B	1	1	0	0	
A AND B	1	0	0	0	NR N NI NC
A OR B	1	1	1	0	OR O OI OC
A FOR B	0	1	1	0	XR X XI XC

2.13 . التعليمات المنطقية

التأثرات هي :

- مرصفتان عامتان (شكل RR) : التعليمات NR ، OR ، XR ،
 - مرصفتان وكلية - ذاكرة (شكل RX) : التعليمات N ، O ، X ،
 - بايته موجودة في التعليمة وبايته موجودة في الذاكرة (الشكل SI عنونة مباشرة) : التعليمات NI ، OI ، XI ،
 - سلسلتان من البايتات في الذاكرة (شكل SS) : التعليمات NC ، OC ، IC .
- تُوضع النتيجة دائماً في المتأثر 1.
- يتم تركيز كود الشرط حسب الطريقة التالية :

CC	
0	إذا كانت النتيجة تعادل صفر
1	إذا كانت النتيجة مختلفة عن صفر

عمليات الإنقطاع الممكنة تتعلّق ، كالعادة ، بمسألة العنونة : تعدّ على المنطقة المخصّصة من الذاكرة ، تعدّ على المنطقة الممكنة من الذاكرة أو مشكلة الزيادة في مضمون المرافص المزدوجة .

التقاطع « و » (AND)

RR COP=14 AND
 $R_1 \leftarrow \text{And} R_2 \rightarrow R_1$
 تتم العملية على 4 بايتات .

RX COP=54 AND
 $R_1 \leftarrow \text{And} (S_2) \rightarrow R_1$
 تتم العملية على أربع بايتات .

SI COP=94 AND
 $(S_1) \leftarrow \text{And} I_2 \rightarrow (S_1)$
 I_2 هي قيمة تلقائية موجودة في التعليمة . العملية تتم على بايتة واحدة .

SS COP=D4 AND
 $(S_1) \leftarrow \text{And} (S_2) \rightarrow (S_1)$

العملية تتم بين منطقتين من الذاكرة بطول مشترك هو I_2 بايتة . ونجري العملية بايتة بعد بايتة من اليسار إلى اليمين . كل شيء يسير كما لو كانت كل بايتة محسوبة وتخزّنة في الذاكرة قبل العبور إلى البايّة التالية .

تطبيق عملي :

تصغير إحدى البتات .

الجمع « أو »

RR COP=18 OR
 $R_1 \leftarrow \text{OR} R_2 \rightarrow R_1$
 تتم على أربع بايتات

RX COP=56 OR
 $(S_2) \leftarrow \text{OR} R_1 \rightarrow R_1$
 تتم على أربع بايتات .

$$OI \quad D_1(B_1), I_2 \quad SI \quad COP=96 \quad OR \quad (S_1) \oplus OU \rightarrow I_2 \rightarrow (S_1)$$

I_2 هي قيمة موجودة في التعلية . تجري العملية على بايتة واحدة .

$$OC \quad D_1(L, B_1), D_2(B_2) \quad SS \quad COP=D6 \quad OR \quad (S_2) \oplus OR \rightarrow (S_1)$$

تتم العملية على منطقتين من الذاكرة بطول مشترك هو L بايتة . وتتم بايتة بعد أخرى من اليسار إلى اليمين .

تطبيق عملي :

جعل إحدى البتات تعادل 1 .

(أو المختصرة « EOR »)

$$XR \quad R_1, R_2 \quad RR \quad COP=17 \quad EXCLUSIVE \ OR \quad R_1 \oplus \text{«EOR»} \rightarrow R_2 \rightarrow R_1$$

تتم العملية على أربع بايتات .

$$X \quad R_1, D_2(X_2, B_2) \quad RX \quad COP=57 \quad EXCLUSIVE \ OR \quad R_1 \oplus \text{«EOR»} \rightarrow (S_2) \rightarrow R_1$$

تتم العملية على أربع بايتات .

$$XI \quad D_1(B_1), I_2 \quad SI \quad COP=97 \quad EXCLUSIVE \ OR \quad (S_1) \oplus \text{«EOR»} \rightarrow I_2 \rightarrow (S_1)$$

I_2 هي قيمة تلفائية موجودة في التعلية . تتم العملية على بايتة واحدة .

$$XC \quad D_1(L, B_1), D_2(B_2) \quad SS \quad COP=D7 \quad EXCLUSIVE \ OR \quad (S_1) \oplus \text{«EOR»} \rightarrow (S_2) \rightarrow (S_1)$$

تجري العملية على منطقتين من الذاكرة بطول مشترك L بايتة ، وتجري بايتة بعد بايتة من اليسار إلى اليمين كما لو كانت كل بايتة قد جرى حسابها وتخزينها في الذاكرة قبل العبور إلى البايته التالية .

تطبيق عملي :

عكس البتة ، مكمل منطقي ، تصفير منطقة من الذاكرة .

3.13 . المقارنات المنطقية

كما في جميع العمليات المنطقية تجري معالجة جميع البتات بنفس الطريقة . لا وجود لأي تمييز للبتة ذات الوزن الأعلى . تتم المقارنة من اليسار إلى اليمين وتتوقف عند أول معادلة . يُركز كود الشرط حسب الطريقة التالية :

(نذكر أن المتأثر الأول هو ذلك الذي يتم بلوغه في التعلية بواسطة المؤشر 1 .

الإقطاعات الممكنة هي تلك المتعلقة بالعنونة وتلك المتعلقة بحدود الكلمات .

0	إذا كانت التأثيرات متساوية
1	إذا كان التأثير الأول أصغر من التأثير الثاني
2	إذا كان التأثير الأول أكبر من التأثير الثاني
3	غير مستعمل

CLR R_1, R_2 RR COP=15 COMPARE LOGICAL

مقارنة بين كامل المرافف .

CL $R_1, D_2(X_2, B_2)$ RX COP=55 COMPARE LOGICAL

مقارنة على أربع بايتات .

CLI $D_1(B_1), I_2$ SI COP=95 COMPARE LOGICAL

مقارنة منطقية مباشرة بين القيمة I_2 الموجودة في التعليمات و (S_1) .

CLC $D_1(L, B_1), D_2(B_2)$ SS COP=D5 COMPARE LOGICAL

مقارنة بين سلاسل تمتد حتى 256 بايت بطول مشترك L .

CLM $R_1, M_3, D_2(B_2)$ RS COP=BD COMPARE LOGICAL CHARACTERS UNDER MASK (370)

القناع M_3 ، المكون من أربع بتات يختار في R_1 من 0 إلى 4 بايتات تقارن بالبايتات المتتالية إنطلاقاً من العنوان S_2 . البتة الأولى من القناع ، إذا كانت معادلة لـ 1 تختار البايتة الأولى من R_1 وهكذا دواليك . يتم تركيز CC .

القناع المعادل لـ 1011 يختار البايتات 0 ، 2 ، 3 من R_1 التي تتم مقارنتها مع ثلاث بايتات إنطلاقاً من S_2 . للمقارنة تتم من اليسار إلى اليمين .

CLCL R_1, R_2 RR COP=OF COMPARE LOGICAL LONG (370)

مقارنة بين سلسلتين من البايتات حيث المتأولين والأطوال موجودة في المرافف الزوجية حسب الإنفاق التالي :



تجري العملية من اليسار إلى اليمين من خلال المتناوين 1 و 2 . إذا لم يكن طول السلسلتين متعادلاً ، يُفترض بأن يكمل الأقصر من اليمين بالسمة «padding» (سمة الحشو) .

العملية تتم بائنة بعد بائنة مع زيادة عناوين وتقصير الطول . وهي قابلة للانقطاع بين مقارنة بائتين . وتتوقف عند أول لا معادلة نلتقيها أو في نهاية السلسلة مع تركيز كود - الشرط .

4.13 . مقارنات منطقية خاصة

لقد قمنا هنا بتصنيف التعليمات التي ، زيادة عن وظيفتها في المقارنة ، تتمتع بعمل خاص . هذه التعليمات تركز كود الشرط بصورة مختلفة .

CS R₁,R₃,D₃(B₂) RS COP=BA COMPARE AND SWAP
(370)

مقارنة بين R₁ و (S₂)

إذا : (S₂) = R₁ عندئذ (S₂) → R₁ و CC → 0 .
إذا : (S₂) ≠ R₁ عندئذ (S₂) → R₁ و CC → 1 .

CDS R₁,R₃,D₃(B₂) RS COP=BB COMPARE DOUBLE AND SWAP
(370)

مقارنة بين R₁ و (S₂)

إذا : (S₂) = R₁ عندئذ (S₂) → R₁ و CC → 0 .
إذا : (S₂) ≠ R₁ عندئذ (S₂) → R₁ و CC → 1 .

للمقارنة CDS تتم حل 64 بتة . وبالنسبة فإن R₁ و R₃ هما مرصفتان مزدوجتان (فقرة 1.10) و S₂ هو عنوان كلمة مزدوجة من الذاكرة .

تُستعمل هاتين التعليمتين لتنفيذ الزمانة بين مهمتين تقسمان منطقة مشتركة من الذاكرة . عندما تتم المعادلة ، فإن كل بلوغ للعنوان S₂ هو ممنوع لأي مُعالج مركزي حتى نهاية عملية النقل (S₂) → R₃ .

TM D₁(B₁),I₂ SI COP=91 TEST UNDER MASK

TM تقوم بانتخاب حالة البتات من البائنة ذات العنوان I₂.S₁ هي قناع من 8 بتات ، كل «1» ، موجود في القناع يسمح بانتخاب وجود بتة «1» في الموقع المناسب من البائنة S₁ .

مثلاً : القناع X'60° أي B'01100000° يفحص وجود «1» في الموقعين 1 و 2 من البائنة . ويجري إهمال المواقع الأخرى . وفي الإجمال ، فإن TM يقوم بتنفيذ عملية AND منطقية بين البائنة التي تم فحصها والقناع دون تعديل البائنة ولكن بتركيز كود الشرط فقط :

0 = CC : جميع البتات التي جرى إختبارها هي 0 أو القناع هو في صفر ،
1 = CC : بعض البتات هي صفر ، وأخرى هي 1 ،
2 = CC : غير مستعمل
3 = CC : جميع البتات المختبرة هي 1 .

11001110	11001110	11001110	البايته المختبرة
00110000	11001000	01011100	القناع
--00----	11--1---	-1-011--	AND
0	3	1	CC

تطبيق :

TM يبدو وكأنه ينتمي إلى CL1 . فعلاً فإن TM يُعتمد لاختبار البتات أكثر من البايتات . مثلاً ، لمعرفة ما إذا كانت البايته هي رقمية نستعمل CL1 لأن القيمة يجب أن تكون محصورة بين F0 و F9 .

TM يمكن أن تُستعمل لتنفيذ تأشير معتمد .

تمارين :

نذكر أن الدالة «AND» تسمح بجعل البتات تعادل صفرأ ، وإن الدالة «OR» تسمح بجعلها 1 وإن «EOR» تسمح بعكسها .

تمرين 1.13 . ضع في صفر ثنائي منطقة بطول $L \geq 256$ بايته ، مرصفاً ، بايته .

تمرين 2.13 . اكتب التعليمة التي تسمح بتركيز قيمة كود الطول في تعليمة من نوع SS .

تمرين 3.13 . بدّل مضمّنون منطقيتين من الذاكرة ، مرصفين ، ريعيين من البتات من نفس البايته .

تمرين 4.13 . تعرّف ما إذا كانت منطقة من الذاكرة مملوءة بفرأغ أو بصفر ثنائي

تمرين 5.13 . قم بإجراء تأشير يؤدي إلى تفريع مرّة على اثنتين بواسطة تحويل منطقة قناع تعليمة BC 15,... إلى BC 0,... .

تمرين 6.13 . قم بإجراء تأشير يؤدي إلى تفريع إلى جميع نقاط العبور ما عدا الأول .

تمرين 7.13 . بدّل جميع أصفار السنار (X'F0') في عدد عشري بفرأغات (X'40') .

تمرين 8.13 . البايته تسمح بتجميع حتى ثمانية مؤشرات ثنائية . لتأخذ البايته INDIC

التي تجمع المؤشرات الثنائية INDEC , INDEC , INDWAIT ,

النامنية على التوالي للقيم السادس عشرية '80'X , '40'X و '20'X من

INDIC (تحتل المؤشرات البتات 0 ، 1 و 2 من INDIC) . اكتب

التعليقات التي تسمح :

- بتعريف INDIC , INDEC , INDEC , INDWAIT ؛

- بتركيز INDWAIT في 1 ؛

- بتركيز INDEC و INDWAIT في 1 ؛

- بتركيز INDEC و INDEC في صفر ؛

- بتفريع إلى ALPHA إذا كانت INDWAIT في «1» ؛
 - بتفريع إلى BETA إذا كانت INDWAIT و INDLEC في «1» ؛
 - بتفريع إلى GAMMA عندما يكون فقط INDLEC أو INDWAIT في «1» ؛
 - بتفريع إلى DELTA عندما تكون INDWAIT و INDLEC في صفر .
- لنفترض بأننا نرغب بربط INDLEC بالبتة 7 من INDIC بدلاً من البتة 0 ، مما يتناسب مع 'X'01' بدلاً من 'X'80' . الحل الخاص بكم هل يسمح بعدم تعديل تعليقات التركيز والاختبار لـ INDLEC ؟

14 . عمليات الإزاحة (Shift)

1.14 . العمليات « المنطقية » والعمليات « الحسابية »

عند دراسة تعليقات الجمع بفاصلة ثابتة ، لاحظنا ، أنه إلى جانب التعليقات A ، AR و AH ، تأتي عمليات الجمع المنطقية . الفرق بين هذين النوعين من العمليات هو التالي :

- تميّز العمليات الجبرية البتة 0 ، المعتبرة كإشارة ، تجري العملية على 31 بتة مع مُرحّل محتمل إلى بتة الإشارة . يجري اختيار الإشارة ويمكن أن تؤدي إلى إنقطاع من نوع overflow .

- العمليات من نوع منطقي لا تأخذ بعين الاعتبار أي تمييز للبتة ذات الوزن الأكبر . تجري معالجة جميع البتات بنفس الطريقة . أي ترحيل في نهاية البتة ذات الوزن الأكبر لا يؤدي إلى انقطاع .

الإزاحة هي عبارة عن نقل إلى اليسار أو إلى اليمين لعدد n من المواقع لتشكيلة ثنائية موجودة في مرصف بسيط (إزاحة بسيطة) أو في مرصف مزدوج (إزاحة مزدوجة) .

عند الإزاحة تضعيب البتات المطروقة . والبتات الداخلة لجهة اليمين هي دائماً صفر . أما البتات التي تدخل من اليسار فيمكن أن تكون إما «0» (إزاحة منطقية إلى اليمين أو إزاحة حسابية إلى اليمين لعدد إيجابي) أو «1» (إزاحة جبرية إلى اليمين لعدد سلمي) . سنرى السبب لاحقاً .

2.14 . الإزاحة الجبرية

تجري الإزاحة الجبرية على القيمة ، أي على 31 بتة (إزاحة بسيطة) أو على 63 بتة (إزاحة مزدوجة) .

- الإزاحة إلى اليمين تؤدي إلى إدخال بتات معادلة لبتة الإشارة .

- الإزاحة إلى اليسار تؤدي إلى إدخال 0. إذا جرى تعديل بته الإشارة سيحدث انقطاع من نوع overflow بفاصلة ثابتة .

الإزاحة الجبرية تؤدي إلى تركيز كود الشرط على الشكل التالي :

CC = 0	إذا كانت النتيجة صفراً .
CC = 1	إذا كانت النتيجة سلبية .
CC = 2	إذا كانت النتيجة إيجابية
CC = 3	إذا كان يوجد overflow (تعديل في بته الإشارة في حالة إزاحة إلى اليسار) .

أمثلة :

لتبسيط العرض سنفترض أن حجم المصرف يعادل ثمان بتات . البته ذات الوزن الأكبر هي إذا بته الإشارة .

مرصف بسيط	مرصف مزدوج
قبل الإزاحة S 00001111 = +15	00000000 11001111 = +207 S
بعد الإزاحة لجهة اليسار ثلاثة S 01111000 = +120 CC = 2	00000110 01111000 = +1656
بعد الإزاحة لجهة اليمين ثلاثة S 00000001 = +1 CC = 2	00000000 00011001 = +25 S
قبل الإزاحة S 11100101 = -27	
بعد الإزاحة لجهة اليمين 1 S 11110010 = -14 CC = 1	
بعد الإزاحة اليسار 4 S 01010000 = +80 CC = 3 OVERFLOW	

3.14 . الإزاحة المنطقية

تعالج الإزاحة المنطقية 32 بتة (إزاحة بسيطة) أو 64 بتة (إزاحة مزدوجة) دون أخذ بالاعتبار البتة ذات الوزن الأكبر . البتات الداخلة هي دائماً «0» . لا يحدث إنقطاع من نوع overflow . لا يجري تعديل في CC .
أمثلة : على ثمان بتات .

10011100

قبل الإزاحة

01110000

بعد الإزاحة لجهة اليسار 2

00100111

بعد الإزاحة لجهة اليمين 2

4.14 . تعليمات الإزاحة

يوجد أربع عمليات إزاحة جبرية ، أربع تعليمات إزاحة منطقية ، وتعليمات إزاحة لعدد عشري . نذكر هذه الأخيرة عند دراسة الحساب العشري .
الإزاحة الجبرية :

SLA	$R_1, D_2(B_2)$	RS	COP=8B	SHIFT LEFT SINGLE	إزاحة بسيطة إلى اليسار
SLDA	$R_1, D_2(B_2)$	RS	COP=8F	SHIFT LEFT DOUBLE	إزاحة مزدوجة إلى اليمين
SRA	$R_1, D_2(B_2)$	RS	COP=8A	SHIFT RIGHT SINGLE	إزاحة بسيطة إلى اليمين
SRDA	$R_1, D_2(B_2)$	RS	COP=8E	SHIFT RIGHT DOUBLE	إزاحة مزدوجة إلى اليمين

الإزاحة المنطقية

SLL	$R_1, D_2(B_2)$	RS	COP=89	SHIFT LEFT SINGLE LOGICAL	إزاحة بسيطة منطقية إلى اليسار
SLDL	$R_1, D_2(B_2)$	RS	COP=8D	SHIFT LEFT DOUBLE LOGICAL	إزاحة منطقية مزدوجة إلى اليسار

SRL R₁D₂(B₂) RS COP=88 SHIFT RIGHT SINGLE LOGICAL
إزاحة بسيطة منطقية إلى اليمين

SRDL R₁D₂(B₂) RS COP=8C SHIFT RIGHT DOUBLE LOGICAL
إزاحة مزدوجة منطقية إلى اليمين .

قواعد مشتركة للإزاحات المنطقية والجبرية

- تتم عمليات الإزاحة على مضمون المرصف R_i .
- بالنسبة لعمليات الإزاحة المزدوجة ، فإن R_i يجب أن يكون مرصفاً مزدوجاً حسب الاتفاق العادي (فقرة 2.10) .
- المتأثر الثاني D₂(B₂) ليس عنواناً :

1- إذا كان B₂ هو المرصف 0 ، فإن البتات الست ذات الوزن الأضعف للنقطة تعطي عدد المواقع المطلوب إزاحتها . SLA 5,3(0) أو SLA 5,3(0) هما عمليتا إزاحة لجهة اليسار لثلاثة مواقع ثنائية .

2- إذا لم يكن B₂ هو المرصف 0 ، فإن المرصف المذكور يحتوي على عدد المواقع المطلوب إزاحتها . ونحصل على الإزاحة بشكل غير مباشر . SRDL 6,0(5) يزحل منطقياً المرصف للزدوج (المرصفان 6 و 7) لعدد المواقع المشار إليها في المرصف 5 .

- وحدها عمليات الإزاحة الجبرية تقوم بتركيز كود الشرط CC حسب اتفاق الفقرة 2.14 .

تمارين :

- تمرين 1.14 - ضع في صفر مرصفاً بواسطة الإزاحة .
- تمرين 2.14 - أضرب واقسم عدداً موجوداً في مرصف على قوة لـ 2 بواسطة الإزاحة . إفحص ، بالنسبة للقسمة ، اتجاه التقريب .
- تمرين 3.14 - إفحص فيما إذا كان زوج من المراصف مزدوج / مفرد هو صفر .
- تمرين 4.14 - برمج إزاحة دائرية لمرصف بسيط .

15 . مسائل

1.15 . الفرز

يتعلّق ذلك بترتيب جدول من الكلمات التي تحتوي على أعداد بفاصلة ثابتة بترتيب تصاعدي . لقد قمنا باختيار الخوارزم الكلاسيكي الذي يُعرف بـ « طريقة الفقاعة » . تقوم الطريقة على فحص عناصر الجدول من اليسار إلى اليمين مع تبديل العناصر المتتالية الموجودة بشكل عشوائي . نضع إلى اليمين العنصر الأكبر كما نلاحظ من المثل التالي :

5	1	3	2
1	5	3	2
1	3	5	2
1	3	2	5

إذا كان N هو حجم الجدول ، نبدأ العملية باعتبار الجدول الثنائي بالحجم $N-1$ وهكذا دواليك ، طالما يوجد عملية تبديل واحدة على الأقل خلال التكرار السابق .

ولو افترضنا أنه خلال فحص الأعداد ، لم نجر أية عملية تبديل فمعنى ذلك إن الترتيب قد حصل .

البرنامج مؤلف من حقتين BCL1 و BCL2 متداخلتين . الحلقة الداخلية BCL2 تفحص الجدول باستعمال مرصف مؤشر PTR : (PTR) هو عنوان العنصر . العناصر التي جرت مقارنتها هي إذا ((PTR)) و ((PTR)+4)) . يتم إنشاء الحلقة بواسطة BXLE . المرصف المزدوج INCR/REFER . يحتوي على الزيادة 4 والحّد $4 * (N-1) + TAB$.

عند إجراء تبديل نقوم بتركيز البايته INDIC في 1 . الحلقة BCL1 تُكرّر BCL2 طالما إن $INDIC=1$.

(1) نذكّر بأنه حسب الترميز للمحدد ، (PTR) يُقرأ « مضمون PTR » وهنا هو إذن عبارة عن عنوان . مضمون هذا العنوان ، أي العنصر المطلوب ، يُرمز إليه بـ ((PTR)) .

LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT
700000				1	TRI	SELECT
700001				2	PIR	FROM
700002				3	INDIC	WHERE
700003				4	REF	GROUP
700004				5	ORDER	ORDER
700005				6	SWAP	ORDER
700006				7	PROD	ORDER
700007				8	USING	ORDER
700008				9	TABLE	ORDER
700009				10	TABLE	ORDER
700010				11	TABLE	ORDER
700011				12	TABLE	ORDER
700012				13	TABLE	ORDER
700013				14	TABLE	ORDER
700014				15	TABLE	ORDER
700015				16	TABLE	ORDER
700016				17	TABLE	ORDER
700017				18	TABLE	ORDER
700018				19	TABLE	ORDER
700019				20	TABLE	ORDER
700020				21	TABLE	ORDER
700021				22	TABLE	ORDER
700022				23	TABLE	ORDER
700023				24	TABLE	ORDER
700024				25	TABLE	ORDER
700025				26	TABLE	ORDER
700026				27	TABLE	ORDER
700027				28	TABLE	ORDER
700028				29	TABLE	ORDER
700029				30	TABLE	ORDER
700030				31	TABLE	ORDER
700031				32	TABLE	ORDER
700032				33	TABLE	ORDER
700033				34	TABLE	ORDER
700034				35	TABLE	ORDER
700035				36	TABLE	ORDER
700036				37	TABLE	ORDER
700037				38	TABLE	ORDER
700038				39	TABLE	ORDER
700039				40	TABLE	ORDER
700040				41	TABLE	ORDER
700041				42	TABLE	ORDER
700042				43	TABLE	ORDER
700043				44	TABLE	ORDER
700044				45	TABLE	ORDER
700045				46	TABLE	ORDER
700046				47	TABLE	ORDER
700047				48	TABLE	ORDER
700048				49	TABLE	ORDER
700049				50	TABLE	ORDER
700050				51	TABLE	ORDER
700051				52	TABLE	ORDER
700052				53	TABLE	ORDER
700053				54	TABLE	ORDER
700054				55	TABLE	ORDER
700055				56	TABLE	ORDER
700056				57	TABLE	ORDER
700057				58	TABLE	ORDER
700058				59	TABLE	ORDER
700059				60	TABLE	ORDER
700060				61	TABLE	ORDER
700061				62	TABLE	ORDER
700062				63	TABLE	ORDER
700063				64	TABLE	ORDER
700064				65	TABLE	ORDER
700065				66	TABLE	ORDER
700066				67	TABLE	ORDER
700067				68	TABLE	ORDER
700068				69	TABLE	ORDER
700069				70	TABLE	ORDER
700070				71	TABLE	ORDER
700071				72	TABLE	ORDER
700072				73	TABLE	ORDER
700073				74	TABLE	ORDER
700074				75	TABLE	ORDER
700075				76	TABLE	ORDER
700076				77	TABLE	ORDER
700077				78	TABLE	ORDER
700078				79	TABLE	ORDER
700079				80	TABLE	ORDER
700080				81	TABLE	ORDER
700081				82	TABLE	ORDER
700082				83	TABLE	ORDER
700083				84	TABLE	ORDER

قبل الفرز

PSW AT ENTRY TO SNAP 07BD1000 0008705E ILC 2 INTC 0033

REGS AT ENTRY TO SNAP

FLTR 0-6 0000000000000000 0000000000000000
REGS 0-7 000011A0 0008703E 00084F54 00087010
REGS 8-15 00000000 00084E20 00084F50 00087000

-STORAGE

087000 00000018 00087140 0A13A140 0004A1B0
087020 00000018 00087140 00087030 00000000
087040 00000018 00087140 00087030 00000000
087060 00000018 00087140 00087030 00000000
087080 00000018 00087140 00087030 00000000
087100 00000018 00087140 00087030 00000000
087120 00000018 00087140 00087030 00000000
087140 00000018 00087140 00087030 00000000

00000000000000000000 00000000000000000000
00000004 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E

00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E

00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E

بعد الفرز

PSW AT ENTRY TO SNAP 07BD1000 0008705E ILC 2 INTC 0033

REGS AT ENTRY TO SNAP

FLTR 0-6 0000000000000000 0000000000000000
REGS 0-7 00000000 0008703E 0008703E 0008703E
REGS 8-15 00000000 00084E20 00084F50 00087000

-STORAGE

087000 00000018 00087140 0A13A140 0004A1B0
087020 00000018 00087140 00087030 00000000
087040 00000018 00087140 00087030 00000000
087060 00000018 00087140 00087030 00000000
087080 00000018 00087140 00087030 00000000
087100 00000018 00087140 00087030 00000000
087120 00000018 00087140 00087030 00000000
087140 00000018 00087140 00087030 00000000

00000000000000000000 00000000000000000000
00000004 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E

00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E

00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E
00000000 0008703E 0008703E 0008703E



من الممكن أن تكتب الخوارزم على الشكل التالي :

(← هورمز التخصيص)

```

INCRE ← 4
INDIC ← 1
REFER ← TAB+(N-1)*4
BCL1: TANT QUE INDIC ≠ 0 FAIRE
      INDIC ← 0
      REFER ← REFER - INCRE
      BCL2: -----
            exploration
            -----
      FIN BCL2
FIN BCL1

```

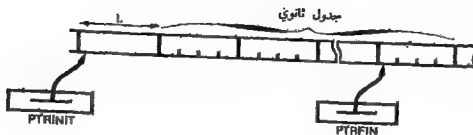
SNAP هي عبارة عن ماكرو تعليمية نموذجية تسمح بالحصول على صورة سادس عشرية من الذاكرة . إستعمالها يتطلب فتح السجل (OPEN) ، إغلاق (CLOSE) ووصف السجل بواسطة الماكرو تعليمية DCB.PRINT NOGEN. (سطر 2) تسمح بإلغاء توليد كود الماكرو تعليمات .

2.15 . إستشارة فرقانية للجدول

يقوم البرنامج على البحث عن وجود أو غياب معلومة من داخل أحد الجداول . البحث المتسلسل يبدو صعباً ويستهلك كثيراً من الوقت عندما يصبح حجم الجدول كبيراً . من الممكن أن نستعمل طريقة الفرقان عندما تكون العناصر منظمة . والصيغة هي التالية :

لنفترض جدولاً TAB من N عنصر منظم نبحث فيه عن موقع المعلومة الموجودة في MOT . نقوم باستشارة العنصر الموجود في وسط TAB ونقارنه بـ MOT . البحث ينتهي عندما نجد التعادل . وإلا نُعيد الكرة ونتابع الاستشارة باختيار واحد من الجدولين الثانويين المشكّلين بواسطة القسمة السابقة حسب موقع العنصر الذي نبحث عنه بالنسبة للعنصر الوسط . بعد كل إستشارة تضيق الفسحة التي نبحث فيها إلى النصف .

سنفترض إن طول العنصر هو L وهذا الطول يعادل قوة (أس) P للعدد 2 ($L=2^P$) . هذا سيسمح بإجراء عمليات ضرب وقسمة بواسطة الإزاحة . سنستعمل مرادف مؤشرات لبلوغ العناصر PTRINIT. سيحتوي على عنوان العنصر الأول من الجدول الثانوي ناقص PTRFIN. سيحتوي على عنوان العنصر الأخير من الجدول الثانوي .



$$\frac{PTRFIN - PTRINIT}{L} : \text{عدد العناصر هو إذاً :}$$

عنوان المتصر الوسط هو :

$$\text{عنوان البداية} + \frac{1}{2} \times \text{عدد العناصر} \times L$$

$$\text{أي : } L + \frac{1}{2} \left(\frac{PTRFIN - PTRINIT}{L} \right) \times L$$

عند القسمة على L يجب إهمال الباقي الذي قد يظهر .

البرنامج التالي مجرى اختباره بعد إجراء نداء لبرنامجين ثانويين مكتوبين بلغة فورتران : LIRE و ECR . وجود نداءات بلغة فورتران من خلال برنامج رئيسي بلغة الموزول يتطلب كتابة التعليقات 59 و 60 غير الموجودة إذن إلا لأسباب توافقية بإشراف النظام المستعمل (FORTRAN G, OS-VS2) .

LDC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
000000				12	DICHO	START 0
				13		
				14		
				15		
				16		
				17		
				18		
				19		
				20		
				21		
				22		
				23		
				24		
				25		
				26		
				27		
				28		
				29		
				30		
				31		
				32		
				33		
				34		
				35		
				36		
				37		
				38		
				39		
				40		
				41		
				42		
				43		
				44		
				45		
				46		
				47		
				48		
				49		
				50		
				51		
				52		
				53		
				54		
				55		
				56		
				57		
				58		
				59		
				60		
				61		
				62		
				63		
				64		
				65		
				66		
				67		
				68		
				69		
				70		
				71		
				72		
				73		
				74		
				75		
				76		
				77		
				78		
				79		
				80		
				81		
				82		
				83		
				84		
				85		
				86		
				87		
				88		
				89		
				90		
				91		
				92		
				93		
				94		
				95		
				96		
				97		
				98		
				99		
				100		
				101		
				102		
				103		
				104		
				105		
				106		
				107		
				108		
				109		
				110		
				111		
				112		
				113		
				114		
				115		
				116		
				117		
				118		
				119		
				120		
				121		
				122		
				123		
				124		
				125		
				126		
				127		
				128		
				129		
				130		
				131		
				132		
				133		
				134		
				135		
				136		
				137		
				138		
				139		
				140		
				141		
				142		
				143		
				144		
				145		
				146		
				147		
				148		
				149		
				150		
				151		
				152		
				153		
				154		
				155		
				156		
				157		
				158		
				159		
				160		
				161		
				162		
				163		
				164		
				165		
				166		
				167		
				168		
				169		
				170		
				171		
				172		
				173		
				174		
				175		
				176		
				177		
				178		
				179		
				180		
				181		
				182		
				183		
				184		
				185		
				186		
				187		
				188		
				189		
				190		
				191		
				192		
				193		
				194		
				195		
				196		
				197		
				198		
				199		
				200		

LOC	OBJECT CODE	ADDR1	ADDR2	SYMT	SOURCE	STATEMENT	
0000DE	1953	000EA		117 IMP	OR	GM	ELEM < (NOT)
0000DE	1970	000EA		118	OR	PTINIT,PTRELEM	
0000E0	1970	000EA		119	OR	TESTFIN	
0000E4	1223	00126		120 SUP	OR	GM	ELEM > (NOT)
0000E4	1220	00126		121	OR	PTINIT,PTRELEM	
0000E4	1220	00126		122	OR	PTREIN,LENG	
0000EA	1982	000F4		124 TESTFIN	OR	GM	
0000EA	1982	000F4		125	OR	PTINIT,PTREIN	
0000EC	1970	000EA		126	OR	PTREIN,LENG	
0000F0	1970	000EA		127	OR	PTREIN,LENG	
0000F4	0703	00130	00130	129 NONTRUCV	OR	GM	PAS TROUVE
0000F4	0703	00130	00130	130	OR	PTREIN,LENG	
000110	1970	0011A	0011A	134	OR	PTREIN,LENG	
000114	1980	000C	000C	147 EPILOGUE	OR	GM	
000116	1980	000C	000C	148	OR	PTREIN,LENG	
000118	1980	000C	000C	149	OR	PTREIN,LENG	
000120	1980	000C	000C	150	OR	PTREIN,LENG	
000122	1980	000C	000C	151	OR	PTREIN,LENG	
000124	1980	000C	000C	152	OR	PTREIN,LENG	
000126	1980	000C	000C	153	OR	PTREIN,LENG	
000128	1980	000C	000C	154	OR	PTREIN,LENG	
000130	1980	000C	000C	155	OR	PTREIN,LENG	
000132	1980	000C	000C	156	OR	PTREIN,LENG	
000134	1980	000C	000C	157	OR	PTREIN,LENG	
000136	1980	000C	000C	158	OR	PTREIN,LENG	
000138	1980	000C	000C	159	OR	PTREIN,LENG	
000140	1980	000C	000C	160	OR	PTREIN,LENG	
000142	1980	000C	000C	161	OR	PTREIN,LENG	
000144	1980	000C	000C	162	OR	PTREIN,LENG	
000146	1980	000C	000C	163	OR	PTREIN,LENG	
000148	1980	000C	000C	164	OR	PTREIN,LENG	
000150	1980	000C	000C	165	OR	PTREIN,LENG	
000152	1980	000C	000C	166	OR	PTREIN,LENG	
000154	1980	000C	000C	167	OR	PTREIN,LENG	
000156	1980	000C	000C	168	OR	PTREIN,LENG	
000158	1980	000C	000C	169	OR	PTREIN,LENG	
000160	1980	000C	000C	170	OR	PTREIN,LENG	
000162	1980	000C	000C	171	OR	PTREIN,LENG	
000164	1980	000C	000C	172	OR	PTREIN,LENG	
000166	1980	000C	000C	173	OR	PTREIN,LENG	
000168	1980	000C	000C	174	OR	PTREIN,LENG	
000170	1980	000C	000C	175	OR	PTREIN,LENG	
000172	1980	000C	000C	176	OR	PTREIN,LENG	
000174	1980	000C	000C	177	OR	PTREIN,LENG	
000176	1980	000C	000C	178	OR	PTREIN,LENG	
000178	1980	000C	000C	179	OR	PTREIN,LENG	
000180	1980	000C	000C	180	OR	PTREIN,LENG	
000182	1980	000C	000C	181	OR	PTREIN,LENG	
000184	1980	000C	000C	182	OR	PTREIN,LENG	
000186	1980	000C	000C	183	OR	PTREIN,LENG	
000188	1980	000C	000C	184	OR	PTREIN,LENG	
000190	1980	000C	000C	185	OR	PTREIN,LENG	
000192	1980	000C	000C	186	OR	PTREIN,LENG	
000194	1980	000C	000C	187	OR	PTREIN,LENG	
000196	1980	000C	000C	188	OR	PTREIN,LENG	
000198	1980	000C	000C	189	OR	PTREIN,LENG	
000200	1980	000C	000C	190	OR	PTREIN,LENG	
000202	1980	000C	000C	191	OR	PTREIN,LENG	
000204	1980	000C	000C	192	OR	PTREIN,LENG	
000206	1980	000C	000C	193	OR	PTREIN,LENG	
000208	1980	000C	000C	194	OR	PTREIN,LENG	
000210	1980	000C	000C	195	OR	PTREIN,LENG	
000212	1980	000C	000C	196	OR	PTREIN,LENG	
000214	1980	000C	000C	197	OR	PTREIN,LENG	
000216	1980	000C	000C	198	OR	PTREIN,LENG	
000218	1980	000C	000C	199	OR	PTREIN,LENG	
000220	1980	000C	000C	200	OR	PTREIN,LENG	
000222	1980	000C	000C	201	OR	PTREIN,LENG	
000224	1980	000C	000C	202	OR	PTREIN,LENG	
000226	1980	000C	000C	203	OR	PTREIN,LENG	
000228	1980	000C	000C	204	OR	PTREIN,LENG	
000230	1980	000C	000C	205	OR	PTREIN,LENG	
000232	1980	000C	000C	206	OR	PTREIN,LENG	
000234	1980	000C	000C	207	OR	PTREIN,LENG	
000236	1980	000C	000C	208	OR	PTREIN,LENG	
000238	1980	000C	000C	209	OR	PTREIN,LENG	
000240	1980	000C	000C	210	OR	PTREIN,LENG	
000242	1980	000C	000C	211	OR	PTREIN,LENG	
000244	1980	000C	000C	212	OR	PTREIN,LENG	
000246	1980	000C	000C	213	OR	PTREIN,LENG	
000248	1980	000C	000C	214	OR	PTREIN,LENG	
000250	1980	000C	000C	215	OR	PTREIN,LENG	
000252	1980	000C	000C	216	OR	PTREIN,LENG	
000254	1980	000C	000C	217	OR	PTREIN,LENG	
000256	1980	000C	000C	218	OR	PTREIN,LENG	
000258	1980	000C	000C	219	OR	PTREIN,LENG	
000260	1980	000C	000C	220	OR	PTREIN,LENG	
000262	1980	000C	000C	221	OR	PTREIN,LENG	
000264	1980	000C	000C	222	OR	PTREIN,LENG	
000266	1980	000C	000C	223	OR	PTREIN,LENG	
000268	1980	000C	000C	224	OR	PTREIN,LENG	
000270	1980	000C	000C	225	OR	PTREIN,LENG	
000272	1980	000C	000C	226	OR	PTREIN,LENG	
000274	1980	000C	000C	227	OR	PTREIN,LENG	
000276	1980	000C	000C	228	OR	PTREIN,LENG	
000278	1980	000C	000C	229	OR	PTREIN,LENG	
000280	1980	000C	000C	230	OR	PTREIN,LENG	
000282	1980	000C	000C	231	OR	PTREIN,LENG	
000284	1980	000C	000C	232	OR	PTREIN,LENG	
000286	1980	000C	000C	233	OR	PTREIN,LENG	
000288	1980	000C	000C	234	OR	PTREIN,LENG	
000290	1980	000C	000C	235	OR	PTREIN,LENG	
000292	1980	000C	000C	236	OR	PTREIN,LENG	
000294	1980	000C	000C	237	OR	PTREIN,LENG	
000296	1980	000C	000C	238	OR	PTREIN,LENG	
000298	1980	000C	000C	239	OR	PTREIN,LENG	
000300	1980	000C	000C	240	OR	PTREIN,LENG	
000302	1980	000C	000C	241	OR	PTREIN,LENG	
000304	1980	000C	000C	242	OR	PTREIN,LENG	
000306	1980	000C	000C	243	OR	PTREIN,LENG	
000308	1980	000C	000C	244	OR	PTREIN,LENG	
000310	1980	000C	000C	245	OR	PTREIN,LENG	
000312	1980	000C	000C	246	OR	PTREIN,LENG	
000314	1980	000C	000C	247	OR	PTREIN,LENG	
000316	1980	000C	000C	248	OR	PTREIN,LENG	
000318	1980	000C	000C	249	OR	PTREIN,LENG	
000320	1980	000C	000C	250	OR	PTREIN,LENG	
000322	1980	000C	000C	251	OR	PTREIN,LENG	
000324	1980	000C	000C	252	OR	PTREIN,LENG	
000326	1980	000C	000C	253	OR	PTREIN,LENG	
000328	1980	000C	000C	254	OR	PTREIN,LENG	
000330	1980	000C	000C	255	OR	PTREIN,LENG	
000332	1980	000C	000C	256	OR	PTREIN,LENG	
000334	1980	000C	000C	257	OR	PTREIN,LENG	
000336	1980	000C	000C	258	OR	PTREIN,LENG	
000338	1980	000C	000C	259	OR	PTREIN,LENG	
000340	1980	000C	000C	260	OR	PTREIN,LENG	
000342	1980	000C	000C	261	OR	PTREIN,LENG	
000344	1980	000C	000C	262	OR	PTREIN,LENG	
000346	1980	000C	000C	263	OR	PTREIN,LENG	
000348	1980	000C	000C	264	OR	PTREIN,LENG	
000350	1980	000C	000C	265	OR	PTREIN,LENG	
000352	1980	000C	000C	266	OR	PTREIN,LENG	
000354	1980	000C	000C	267	OR	PTREIN,LENG	
000356	1980	000C	000C	268	OR	PTREIN,LENG	
000358	1980	000C	000C	269	OR	PTREIN,LENG	
000360	1980	000C	000C	270	OR	PTREIN,LENG	
000362	1980	000C	000C	271	OR	PTREIN,LENG	
000364	1980	000C	000C	272	OR	PTREIN,LENG	
000366	1980	000C	000C	273	OR	PTREIN,LENG	
000368	1980	000C	000C	274	OR	PTREIN,LENG	
000370	1980	000C	000C	275	OR	PTREIN,LENG	
000372	1980	000C	000C	276	OR	PTREIN,LENG	
000374	1980	000C	000C	277	OR	PTREIN,LENG	
000376	1980	000C	000C	278	OR	PTREIN,LENG	
000378	1980	000C	000C	279	OR	PTREIN,LENG	
000380	1980	000C	000C	280	OR	PTREIN,LENG	
000382	1980	000C	000C	281	OR	PTREIN,LENG	
000384	1980	000C	000C	282	OR	PTREIN,LENG	
000386	1980	000C	000C	283	OR	PTREIN,LENG	
000388							

16 . الحساب العشري

1.16 . عموميات

تقدّم التعليمات الحسابية العشرية وسائل لإجراء الحسابات على الأعداد العشرية « المتراسة packed » التي رأيناها في الفقرة 3.5.2. ج . ولاحقاً سندرس عملية تحويلها لمعطيات .

التعليمات الحسابية هي بنسق SS وتستعمل الطولين L_1 و L_2 للمتأثرين : يبقى طول المتأثرات محدوداً بـ 16 بايتة (31 رقماً عشرياً زائد الإشارة في التمثيل المتراس و 16 رقماً وإشارة في التمثيل للموسّع) لأنها تقسم المنطقة L بالنسق SS . شكل هذه التعليمات هو التالي :

COP	L_1	L_2	B_1	D_1	B_2	D_2
-----	-------	-------	-------	-------	-------	-------

ونشير إلى أنّه جرت العادة بالنسبة للتعليمات SS بأن تكون القيم المؤوّلة في المناطق L هي بالطول المذكور في تعليمة مؤؤل ناقص 1 . هكذا ، فالتعليمة :

A P ALPHA (16), BETA (10)

سيتم تأويلها مع القيم الثنائية 1111 و 1001 بالنسبة للطول .

تضع التعليمات الحسابية النتيجة في المتأثر الأول الذي يتم إلغاؤه ويجب أن يكون هذا المتأثر بطول كافٍ لاستيعاب النتيجة دون حدوث overflow وقطع للعدد . يظهر overflow إذا لم يكن المتأثر الأول بالطول المناسب لاستيعاب النتيجة . عندما تكون $L_1 < L_2$ لا يحدث overflow إذا لم يكن هناك مُرحّل (carry) خارج الإمكانات المقدّمة من الطول L_1 . ويمكن تقنين overflow بواسطة البتة SPM .

عند إجراء العمليات ، فإن الفاصلة لا تُمثّل والتراسف يتم لجهة اليمين ، كما يمكن حصر المتأثرات بواسطة عمليات إزاحة عشرية مناسبة .

تتحقق الدارات ، خلال التنفيذ ، من صلاحية الأرقام العشرية والإشارات .
والتقاء عنصر غير صالح يؤدي إلى انقطاع من نوع استثناء بالمعطيات .

المتأثرات 1 و 2 يمكن أن تندمج بشرط أن تكون بنفس المواقع (متراصفة) بالنسبة
للبيانات ذات الوزن الأضعف . من الممكن هكذا إضافة عدد إلى نفسه :
مثلاً :

ALPHA بعنوان

0	0	0	1	2	3	4	5	6	S
---	---	---	---	---	---	---	---	---	---

التعليمة :

8456 إلى S123456 AP ALPHA(5),ALPHA+3(2)

يتم تركيز كود الشوط CC حسب إشارة النتيجة .

2.16 . التعليقات

AP D₁(L₁,B₁),D₂(L₂,B₂) SS COP=FA ADD DECIMAL
(S₁) + (S₂) + (S₁)

يتم تركيز كود الشرط CC .

ZAP D₁(L₁,B₁),D₂(L₂,B₂) SS COP=FB ZERO AND ADD
(S₂) → (S₁)

تعادل العملية جمع عدد إلى صفر . ويتم تركيز CC .

BP D₁(L₁,B₁),D₂(L₂,B₂) SS COP=FB SUBTRACT DECIMAL
(S₁) - (S₂) + (S₁)

تركيز CC .

MP D₁(L₁,B₁),D₂(L₂,B₂) SS COP=FC MULTIPLY DECIMAL
(S₁) X (S₂) + (S₁)

يجب أن نحصل على : $L_2 \leq 8$ و $L_1 < L_2$ وإلا سيحدث إنقطاع .
CC يبقى بدون تعديل .

DP D₁(L₁,B₁),D₂" . 3) SS COP=FD DIVIDE DECIMAL
(S₁) : (S₂) + (S₁)

يجري وضع النتيجة إلى اليسار في (S₁) . الباقي يُنزّل إلى اليمين في (S₁)
وينفس طول S₂ .

حجم نتيجة القسمة هو 8 بتات : $L_2 - L_1$ يجب أن نحصل على $L_2 \leq 8$
و $L_1 < L_2$ وإلا سيحدث إنقطاع ⁽¹⁾ CC بدون تعديل .

(1) إنباء : يعلّق ذلك بالطول I. بلغة الموزل وليس بطول الرقم .

CP $D_1(L_1, B_1), D_2(L_2, B_2)$ SS COP=F8 COMPARE DECIMAL

عجري مقارنة للتأخرين ويتم تحليل مضمون CC. إذا كانت أطوال
'التأخرات غير متعادلة'، فإن المنطقة الأصغر عجري ملؤها بصفر لجهة
اليسار.

SRP $D_1(L_1, B_1), D_2(B_2), I_1$ SS COP=F0 SHIFT AND ROUND DECIMAL
(370)

يجب الانتباه إلى التنسيق لإخفاص هذه التعليمات. عند التأويل، فإن I_1 تأخذ
الموقع الطبيعي المحفوظ لـ I_1 .

- S_1 هو عنوان التأثير المطلوب لإزاحة.

- I_1 هو الطول.

- $D_2(B_2)$ ليس عنواناً؛ البتات الست ذات الوزن الأضعف والمعتبرة
كمعلم صحيح بإشارة، تدل على اتجاه وعلم الأرقام العشرية المطلوب
إزاحتها. ويجري إسمال البتات الأخرى. القيمة السالبة (مكمّل إلى
2) هي إزاحة إلى اليمين والنتيجة السالبة هي إزاحة إلى اليسار.

- I_1 هو «عامل التلويز» يُستعمل للإزاحات إلى اليمين. تضاف قيمته
إلى الرقم المستخرج بالإزاحة إلى اليمين والمرحّل المحتفل يرتدّ إلى
اليسار.

- توضع النتيجة في (S_1) .

- لا تشترك الإشارة بعملية الإزاحة.

17 الحلب بفاصلة متحركة

لم يبد لنا أساسياً شرح هذه التعليقات بكثير من العناية كما جرى بالنسبة للتعليقات السابقة . فدراسة هذه المجموعة من التعليقات لن نحمل لنا سوى قليلاً من المعلومات الجديدة حول الأوالية الأساسية لتشغيل المكتبات ، بينما نحن نهتم بالدرجة الأولى بهذه الأوالية . ولكن المستعمل الذي فهم جيداً كل ما هو سابق لن ينزعج كثيراً من متابعة هذا الفصل . نفترض هنا بأن القارئ قد استوعب قراءة الفقرة 3.5.2 . ب حول الفاصلة المتحركة في تمثيل المعطيات . ولكي نتذكر بسهولة الكود الحرفي لهذه العمليات ، من الجيد أن نراجع الفقرة 2.10 المتعلقة بالترميز : الحرف النهائي «R» يختص بالتعليمة RR ، والأحرف E ، U ، D ، W و X هي نسق القصير المعايير (normalized) ، والقصير غير المعايير والطويل المعايير والطويل غير المعايير والموسع .

1.17 . عموميات

هذه التعليقات تعمل مع المرافف المتحركة المرقمة 0 ، 2 ، 4 و 6 بطول 64 بتة . الأعداد بفاصلة متحركة القصيرة توضع في الـ 32 بتة ذات الوزن الأكبر من المرافف خلال العمليات . في هذه الحالة فإن الأوزان الضعيفة يجري إسمالها . الأعداد الطويلة بالفاصلة المتحركة تشغل كامل المرافف والأعداد الموسعة بفاصلة متحركة تشغل مرصفين متتاليين . يجري تركيز موقع كود الشرط كالعادة :

جدول 1.17

CC	بالنسبة للتعليقات الجبرية	بالنسبة للمقلونات
0	نتيجة صفر	متأثر 1 = متأثر 2
1	نتيجة سلبية	متأثر 1 > متأثر 2
2	نتيجة إيجابية	متأثر 1 < متأثر 2
3		

2.17 التعليمات

يحدد نفس الخصائص التي رأيناها لدى معالجة الأعداد بفاصلة ثابتة . في حالة الشك بالإمكان مراجعتها

LER R_1, R_2	RR	COP=38	LOAD	متأثرات قصيرة
LE $R_1, D_2(X_2, B_2)$	RX	COP=78	LOAD	متأثرات قصيرة
LDR R_1, R_2	RR	COP=28	LOAD	متأثرات طويلة
LD $R_1, D_2(X_2, B_2)$	RX	COP=68	LOAD	متأثرات طويلة

CC دون تعديل

LTER R_1, R_2	RR	COP=32	LOAD AND TEST	متأثرات قصيرة
LTDR R_1, R_2	RR	COP=22	LOAD AND TEST	متأثرات طويلة
LCER R_1, R_2	RR	COP=33	LOAD COMPLEMENT	متأثرات قصيرة
LCDR R_1, R_2	RR	COP=23	LOAD COMPLEMENT	متأثرات طويلة
LNER R_1, R_2	RR	COP=31	LOAD NEGATIVE	متأثرات قصيرة
LNDR R_1, R_2	RR	COP=21	LOAD NEGATIVE	متأثرات طويلة
LPER R_1, R_2	RR	COP=30	LOAD POSITIVE	متأثرات قصيرة
LPDR R_1, R_2	RR	COP=20	LOAD POSITIVE	متأثرات طويلة

تركيز أو تعديل CC

LRER R_1, R_2 (370)	RR	COP=35	LOAD ROUNDED	التأثر 2 الطويل
LRDR R_1, R_2 (370)	RR	COP=25	LOAD ROUNDED	يجري تدويره ووضع في التأثير الأول القصير
				التأثر الموسع
				يجري تدويره ووضع في التأثير الأول الطويل

دون تعديل

STE $R_1, D_2(X_2, B_2)$	RX	COP=70	STORE	متأثرات قصيرة
STD $R_1, D_2(X_2, B_2)$	RX	COP=80	STORE	متأثرات طويلة

CC دون تعديل

CER R_1, R_2	RR	COP=39	COMPARE	متأثرات قصيرة
CE $R_1, D_2(X_2, B_2)$	RX	COP=79	COMPARE	متأثرات قصيرة
CDR R_1, R_2	RR	COP=29	COMPARE	متأثرات طويلة
CD $R_1, D_2(X_2, B_2)$	RX	COP=69	COMPARE	متأثرات طويلة

تركيز أو تعديل CC

AER R_1, R_2	RR	COP=3A	ADD NORMALIZED	متأثرات قصيرة
AE $R_1, D_2(X_2, B_2)$	RX	COP=7A	ADD NORMALIZED	متأثرات قصيرة
ADR R_1, R_2	RR	COP=2A	ADD NORMALIZED	متأثرات طويلة
AD $R_1, D_2(X_2, B_2)$	RX	COP=6A	ADD NORMALIZED	متأثرات طويلة
AXR R_1, R_2 (370)	RR	COP=36	ADD NORMALIZED	متأثرات موسعة

تركيز أو تعديل CC

AUR R_1, R_2	RR	COP=3E	ADD UNNORMALIZED (op)	متأثرات قصيرة
AU $R_1, D_2(X_2, B_2)$	RX	COP=7E	ADD UNNORMALIZED (o)	متأثرات قصيرة
AWR R_1, R_2	RR	COP=2E	ADD UNNORMALIZED (of)	متأثرات طويلة
AW $R_1, D_2(X_2, B_2)$	RX	COP=6E	ADD UNNORMALIZED (op)	متأثرات طويلة

تركيز أو تعديل CC

SER R_1, R_2	RR	COP=3B	SUBTRACT NORMALIZED	متأثرات قصيرة
SE $R_1, D_2(X_2, B_2)$	RX	COP=7B	SUBTRACT NORMALIZED	متأثرات قصيرة
SDR R_1, R_2	RR	COP=2B	SUBTRACT NORMALIZED	متأثرات طويلة
SD $R_1, D_2(X_2, B_2)$	RX	COP=6B	SUBTRACT NORMALIZED	متأثرات طويلة
SXR R_1, R_2 (370)	RR	COP=27	SUBTRACT NORMALIZED	متأثرات موسعة

تركيز أو تعديل CC

SUR R_1, R_2	RR	COP=3F	SUBTRACT UNNORMALIZED	متأثرات قصيرة
SU $R_1, D_2(X_2, B_2)$	RX	COP=7F	SUBTRACT UNNORMALIZED	متأثرات قصيرة
SWR R_1, R_2	RR	COP=2F	SUBTRACT UNNORMALIZED	متأثرات طويلة
SW $R_1, D_2(X_2, B_2)$	RX	COP=6F	SUBTRACT UNNORMALIZED	متأثرات طويلة

تركيز أو تعديل CC

MER R_1, R_2	RR	COP=3C	MULTIPLY	متأثرات قصيرة ونتيجة موسعة
ME $R_1, D_2(X_2, B_2)$	RX	COP=7C	MULTIPLY	متأثرات قصيرة ونتيجة موسعة
MDR R_1, R_2	RR	COP=2C	MULTIPLY	متأثرات طويلة
MD $R_1, D_2(X_2, B_2)$	RX	COP=6C	MULTIPLY	متأثرات طويلة
MXDR R_1, R_2 (370)	RR	COP=27	MULTIPLY	متأثرات طويلة ونتيجة موسعة
MXD $R_1, D_2(X_2, B_2)$ (370)	RX	COP=67	MULTIPLY	متأثرات طويلة ونتيجة موسعة
MXR R_1, R_2 (370)	RR	COP=26	MULTIPLY	متأثرات موسعة

دون تعديل

DER R_1, R_2	RR	COP=3D	DIVIDE	متأثرات قصيرة
DE $R_1, D_2(X_2, B_2)$	RX	COP=7D	DIVIDE	متأثرات قصيرة
DDR R_1, R_2	RR	COP=2D	DIVIDE	متأثرات طويلة
DD $R_1, D_2(X_2, B_2)$	RX	COP=6D	DIVIDE	متأثرات طويلة

لا تتغير CC

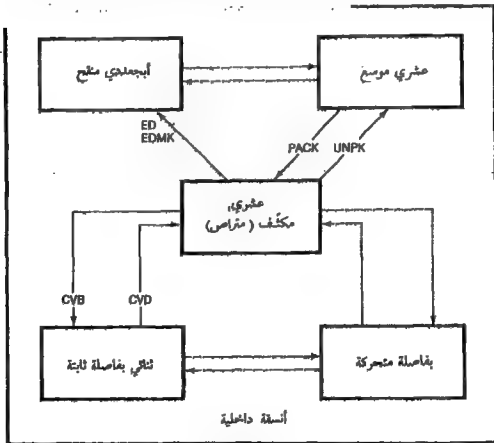
HER R_1, R_2	RR	COP=24	HALVE	متأثرات قصيرة
HDR R_1, R_2	RR	COP=24	HALVE	متأثرات طويلة

يُقسم المتأثر الثاني على 2 وتوضع نتيجة القسمة للعلامة في المتأثر الأول .

18 . تعليمات التحويل والتمثيل

1.18 . عموميات

لقد رأينا أن النظام 370 كان يتمتع بثلاث طبقات من الدارات الحسابية العاملة بثلاث طرق مختلفة لتمثيل المعطيات الرقمية . ولكن ، للمعطيات الداخلة إلى الذاكرة تكون عادة مكدّمة بتمثيل أبجدي . من هنا ، فإن كل عملية حسابية على معطى رقمي داخل إلى المكنة ، من خلال ناقل بطاقات مثلاً ، يمكن أن تتطلب عدة عمليات تحويل للتمثيل قبل معالجته بالحساب العشري ، الثنائي أو بفاصلة متحركة . المخطط 1.18 يعرض مختلف الأشكال الداخلية وعمليات النقل الممكنة التي تتم بواسطة هذه العمليات . الخطوط المنقطة تمثل التحويلات التي تجريها برامج متخصصة .



مخطط 1.18

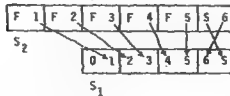
2.18 . تعليمات التحويل

PACK $D_1(L_1, B_1), D_2(L_2, B_2)$ SS COP=F2 PACK
(S_2) \rightarrow (S_1)

عشري مكثف عشري موسع
(متراس)

هذه العملية تحول منطقة S_2 ، يفترض إنها عشرية موسعة ، إلى عشرية متراسة . التحويل يتم من اليمين إلى اليسار بدون تحقق من صلاحية الأكواد .

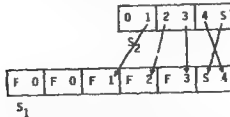
إذا كانت المنطقة S_1 أكبر من الضروري ، فهي تكمل بأصفار (00) لجهة اليسار .
إذا كانت S_1 قصيرة جداً يحدث قطع لجهة اليسار .
 S_1 و S_2 يمكن أن تتراكبا .



UNPK $D_1(L_1, B_1), D_2(L_2, B_2)$ SS COP=F3 UNPACK
(S_2) \rightarrow (S_1)

عشري موسع عشري مكثف

العملية تحول منطقة S_2 ، يفترض إنها عشرية متراسة ، في S_1 عشري موسع .
التحويل يتم من اليمين إلى اليسار ، بدون تحقق من صلاحية الأكواد .
إذا كانت المنطقة S_1 أصغر ، يحدث قطع أو بتر لجهة اليسار .
إذا كانت طويلة تستكمل بأصفار (F0) لجهة اليسار .
 S_1 و S_2 يمكن أن تتراكبا .



CVB $R_1, D_2(X_2, B_2)$

RX COP=4F CONVERT TO BINARY
(S_2) \rightarrow R_1

ثنائي عشري متراس
محصورة في كلمة مزدوجة

صلاحية الاشارة والبيانات الرقمية في S2 يتم التحقق منها . كل خطأ يؤدي إلى انقطاع .
يفترض بأن تكون S2 عبارة عن عنوان لكلمة مزدوجة بطول 8 بايتات .
يُجَدُّ التحويل بالأعداد القصوى والصُّغرى التي من الممكن تمثيلها في 32 بتة ، أي :

$$-2\ 147\ 483\ 648 \leq +2\ 147\ 483\ 647.$$

CVD	R ₁ ,D ₂ (X ₂ ,B ₂)	RX	COP=4E	CONVERT TO DECIMAL
				R ₁ → (S ₂)
			عشري متراس	ثنائي
			موجود في كلمة مزدوجة	
			يتألف العدد العشري الحاصل من 15 رقماً إضافة إلى الإشارة : «C»	
			للجمع (+) و«D» للتناقص (-) . يبقى كود الشرط بدون تغيير .	

3.18 . التنقيح والطباعة

إن مضمون كلمة آلية ثنائية ، مُعطى عشري أو بفاصلة متحركة يجب ، قبل طباعته أن يخضع لتحويل معين . يجب أن يتم تحويل قيمته الثنائية إلى أكواد من السيات القابلة للطباعة . قد يكون من الضروري إدخال فاصلة ، نقطة عشرية ، إشارة أو سيات تعبئة (حالة طباعة الشيكات) .

يوجد تعليمتان ED وEDMK تحقّقان هذا العمل بتحويل منطقة أولية (عشري متراس) إلى منطقة تنقيح وطباعة .
مثلاً :

منطقة أولية

0	0	1	2	3	4	5	0
---	---	---	---	---	---	---	---

منطقة تنقيح

5	C	5	C	5	C	6	0	F	1	F	2	F	3	4	B	F	4	F	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

* * * - 1 2 3 . 4 5

سيات تعبئة إدخال فاصلة عشرية

لكي يتم هذا ، فإن المبرمج يضع في حيز الطباعة قناعاً مؤلفاً من :
- سمة تعبئة .

- أكواد تدل على : مواقع الأرقام ، المكان الذي من خلاله يتم تحويل الأصفار «0» بدون ذات معنى ، السيات المطلوب إدخالها في نهاية سجل الطباعة .

هذه التعليمات تعمل بعلاقة مع مؤشر ثنائي يُدعى « مؤشر معني » . يوضع هذا المؤشر في « 1 » عندما نلتقي برقم ذي معنى في المنطقة الأولية أو عندما نلتقي مكان الأصفار التي من الواجب تحويلها .

تتمتع هنا على العمل الجاري بواسطة « صور » الطباعة بلغة كويول . لن يتم شرح هذه التعليقات هنا وننصح بمراجعة وثائق IBM370.

ED $D_1(L, B_1), D_2(B_2)$ SS COP=DE EDIT

S_1 : منطقة الطباعة ، بطول L ويحتوي حل القناع .
 S_2 : عنوان المنطقة الأولية (المنبع هو منطقة عشرية مترجمة) . يتم تعديل CC حسب إشارة آخر حقل .

EDMK $D_1(L, B_1), D_2(B_2)$ SS COP=DF EDIT AND MARK

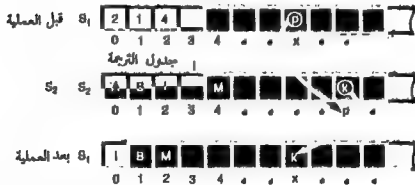
تتمتع S_1 و S_2 بنفس المعنى . عنوان الرقم الأول في المعنى يُؤخذ في المصفى 1 .
 يتم تعديل مضمون CC حسب إشارة آخر حقل .

4.18 . الترجمة

TR $D_1(L, B_1), D_2(B_2)$ SS COP=DC TRANSLATE

ترجمة سلسلة (S_1) بطول L حسب جدول موجود في S_2 بطول أقصى يبلغ 256 بايتة .

قبل العملية ، فإن البايته ($0 \leq X < L$) $S_1 + X$ تحتوي حل الرقم p ($0 \leq p \leq 255$) الذي يستخدم كنقطة إدخال إلى الجدول .
 بعد العملية : $(S_1 + X) + (B_2 + p)$ يبقى CC بدون تعديل .



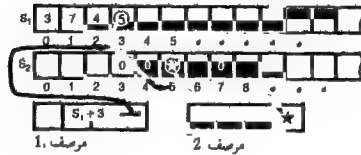
S_1 : منطقة البحث بطول L .

S_2 : عنوان جدول الترجمة .

التعليمة تستعمل المصفين 1 و 2 .

تؤخذ البايته الأولى من المنطقة S_1 بعين الاعتبار . كما في TR ، فإن قيمته النهائية تتكامل نقطة دخول في S_2 .

إذا كانت البايئة المناسبة S_2 مختلفة عن صفر فإن قيمتها تُخزّن في المرفف 2 وعنوان المنطقة التي تسمح بإيجاد التناوب يُخزّن في المرفف 1 .
والأ فإن العملية تاتبع مع البايئة التالية من S_1 . يتم تركيز CC :
 $CC = 0$ إذا كانت المنطقة S_1 قد جرى إستكشافها كلياً وبجميع البايئات التي جرى إستبارها من S_2 كانت صفراً .
 $CC = 1$ إذا جرى إستكشاف S_1 بشكل جزئي ولم تكن البايئة الأخيرة المختارة صفراً .
 $CC = 2$ إذا جرى إستكشاف المنطقة كلياً وكانت البايئة الأخيرة المختارة مختلفة عن صفر .



$CC = 1$
في هذا المثل ، لنستطيع متابعة إستكشاف المنطقة ، يجب إحتداد تعديل لعنوان الانطلاق والطول المستكشف .
 R_{10-25} و R_{10-7} يقفان دون تعديل .
 S_1 لا يتم تعديلها .

تمارين :

تمرين 1.18 - إعادة تنظيم منطقة من الذاكرة .

لنفترض منطقة ARTICLE من 10 بايتات نرغب بنقل البايتات 5 ، 6 ، 7 ، 1 ، 2 إلى المنطقة CLE

ARTICLE	A	B	C	D	E	F	G	H	I	J
CLE	F	G	H	B	C					

اكتب التعليمات التي تسمح بإجراء هذا العمل . في نفس الفكرة نرغب بعكس سلسلة من الـ 5 بايتات . هذا النظام يستعمل لإعادة تنظيم مفاتيح القرص .

تمرين 2.18 - لنفترض منطقة مؤلفة من 8 بايتات بقيم ثنائية موجودة بين 0 و 15 . نرغب باستبدالها بالكود EBCDIC المناسب للقيم السادس عشرية : سيجري إستبدال 0 بواسطة 'C0' ، و 10 بواسطة 'CA' . . . اكتب التعليمات المناسبة .

هذه الأوامر يمكن أن تستعمل ، بعد عملية تحويل بسيطة ، لطباعة مضمون سادس عشري لكلمة من الذاكرة ، للتحضير للطباعة بواسطة DUMP (دلق) .

19 الانقطاع والادخال والاخراج

(Interruptions and I/O)

1.19 . الانقطاعات

لن يكون موضوعنا تفصيل نظام إدارة الانقطاعات هنا ، ولكن فقط إعطاء القارئ إشارات بالنسبة لطبيعة هذه المسألة . لتفصيلات أكثر ننصح بمراجعة وثائق المنشئ Principles of operation .

1.1.19 . صيغة الانقطاعات

الانقطاع هو عبارة عن إشارة كهربائية ، مُرسلة من أحد أعضاء النظام ومعروفة من قِبل الوحدة المركزية . يتيح الانقطاع عن حادثة تتطلب عادةً معالجة مباشرة . لبعض الحوادث صفة خاصة مستعجلة تتطلب تعليق دوران تنفيذ أحد البرامج الجارية كي يتم معالجة الإشارة المرسلة . في النظام IBM 370 ، الحوادث القادرة على تفريع ووقف تنفيذ البرنامج قد جرى تصنيفها حسب أولوية متناقصة :

- نداء للمشرف (call supervisor) ،

- برنامج ،

- عطل في المكنة ،

- إشارة خارجية ،

- عملية إدخال - إخراج (I/O) ،

- إشارة مؤثر (operator signal) .

يرتبط بكل فئة درجة إستعجال معينة . نتكلم هنا عن ستة مستويات من الانقطاعات ونظام معالجة الحوادث يجري حسب الأولوية المعتمدة .

2.1.19 . أوالية الإنقطاع

ندكر بأن المفهوم الذي يدور حوله البرنامج مؤلف من كلمة حالة البرنامج PSW ومن مضمون المرافف العامة والمتحركة المرتبطة به . نشير أيضاً إلى أنه في كل لحظة ، PSW تحتوي على القيمة الحالية لعداد البرنامج . يؤدي تعليق دوران البرنامج أوتوماتيكياً

إلى تخزين مضمون هذه المراسف كي نستطيع معاودة تنفيذ هذا البرنامج المقطوع عند الحاجة . هكذا فالانقطاع يؤدي إلى إطلاق العملية التالية :

1- بشكل أوتوماتيكي (أي بواسطة العتاد (hardware)) ، فإن وصول إشارة الانقطاع تؤدي إلى نسخ PSW الخاصة بالبرنامج الجاري في منطقة محددة من الذاكرة ، تُميز فئة الانقطاع . تدعى هذه الكلمة PSW « الكلمة القديمة » .

2- بشكل أوتوماتيكي ، يأخذ العتاد على عاتقه الكلمة الجديدة PSW الموجودة على عنوان من الذاكرة حسب فئة الانقطاع . منذ هذه اللحظة ، يمكن تنفيذ برنامج جديد : وتبدأ معالجة الانقطاع .

3- بعد الإنتهاء من معالجة الانقطاع ، يمكن معاودة العمل بالبرنامج المقطوع وذلك بواسطة إعادة ترميم الكلمة PSW وإعادة تخزين المراسف بالمعلومات التي كان يحتويها قبل قطع البرنامج .

نضيف أن معالجة الانقطاع يمكن أن تُقطع بدورها بواسطة حادثة أكبر أولوية . مجموعة البرامج التي تعالج الانقطاعات تعتبر جزءاً من نظام التشغيل وتدعى نظام إدارة الانقطاعات .

3.1.19 . قناع الانقطاعات

هذه الأولوية الأساسية يمكن ، ضمن بعض الشروط ، أن يتم « تقنيها » بواسطة المبرمج . بواسطة تفسير الأقنة في الكلمة PSW يمكن للمبرمج أن يمنع أخذ الحوادث الطارئة بالحسبان . هكذا يمكن إهمال الفيض overflow الناتج عن الحساب وذلك بتركيز القناع المناسب بواسطة التعليمات SPM . الانقطاع المبرمج المُقنع لا يتم أبداً ، كما يوضع الانقطاع المُقنع الناتج عن النظام في الانتظار حتى يجري رفع القناع أو القيد عنه . التعليمات SSM التي تسمح بتعديل قناع النظام هي تعليمات خاصة .

4.1.19 . الانقطاعات الناتجة عن البرنامج

سنعطي هنا أسباب الانقطاعات الناتجة عن البرنامج . وهي تولّد عادة بسبب خطأ في البرمجة . وتجري الإشارة إليها بواسطة ظهور كود للعودة OxC يُدعى «completion code» أو كود الانتهاء .

لتفاصيل أكثر يجب على القارئ أن يراجع وثائق IBM الخاصة .

OPERATION EXCEPTION

code = 0C1

يتج هذا الانقطاع عندما يكون هناك محاولة لتنفيذ تعليمية بكود عملية غير صالح .

PRIVILEGED-OPERATION EXCEPTION

code = 0C2

محاولة لتنفيذ تعليمة خاصة بينما تكون المكنة في صيغة المسألة .

EXECUTE EXCEPTION

code = 0C3

التعليمة EX تعود إلى تعليمة أخرى EX .

PROTECTION EXCEPTION

code = 0C4

يتعلق ذلك ببلوغ موقع محمي من الذاكرة .

ADRESSING EXCEPTION

code = 0C5

يتعلق ذلك بمحاولة بلوغ موقع غير موجود في الذاكرة .

SPECIFICATION EXCEPTION

code = 0C6

هذا الانقطاع يغطي أكثر الحالات ، لن نذكر سوى الأكثر شيوعاً . يتعلق ذلك بمسألة الحدود : لا تحصر التعليمة بحدود نصف كلمة أو معطى غير مسطر كما تحتاج التعليمة التي تُرجع إليها .

DATA EXCEPTION

code = 0C7

يتعلق ذلك بمشكلة ناتجة عن تعليمة CVB أو تعليمة عشرية .

FIXED-POINT-OVERFLOW EXCEPTION

code = 0C8

overflow في تمثيل بفاصلة ثابتة .

FIXED POINT DIVIDE EXCEPTION

code = 0C9

يتعلق ذلك بالقسمة على صفر ، أو بنتيجة قسمة يزيد حجمها عن حجم المرصف أو بتحويل إلى ثنائي (CVB) حيث النتيجة تزيد عن 31 بتة .

DECIMAL-OVERFLOW EXCEPTION

code = 0CA

تلتقي هذه التعليمة في عملية على أعداد عشرية ، عندما يتم فقدان البتات ذات الأوزان العليا لأن المنطقة النهائية هي أصغر من أن تحتوي على النتيجة .

DECIMAL-DIVIDE EXCEPTION

code = 0CB

يتعلق ذلك بالقسمة على صفر في عملية بالنظم العشري .

EXPONENT-OVERFLOW EXCEPTION

code = 0CC

الأس الخاص بالنتيجة يزيد عن 127 والقسم العشري (mantisse) ليس صفراً .

EXPONENT-UNDERFLOW EXCEPTION

code = 0CD

الأس هو سلبى والقسم العشري ليس صفراً .

SIGNIFICANCE EXCEPTION

code = 0CE

في عملية جمع أو طرح على أعداد بفاصلة متحركة والقسم العشري هو صفر .

FLOATING POINT-DIVIDE EXCEPTION

code = 0CF

قسمة على صفر لأعداد بفاصلة متحركة .

5.1.19 . تعليمات مرتبطة بالانقطاعاتSPM R₁

RR COP=04

SET PROGRAM MASK

R₁(2-7) → CC,

أقنعة البرنامج



البتات من 2 إلى 7 من المرصف العام R₁ تُخزَّن (البتات 2 و 3) في CC وفي (البتات 4 و 7) قناع البرنامج . نشير هنا إلى أن التعليمات BAL وBALR تشحن المرصف R₁(2-7) بالكود CC وقناع البرنامج .

SVC

RR COP=0A

SUPERVISOR CALL

هذه التعليمة تؤدي إلى انقطاع بكود I . الكلمة القدية PSW تُخزَّن في الذاكرة على العنوان 32 والكلمة الجديدة PSW تؤخذ على العنوان 96 .

MC

D₁(B₁), 1, 2

SI COP=AF

MONITOR CALL

(370)

تطلق برنامج انقطاع عندما تكون بة خاصة من القناع الموجّه في 1 .

STCK

D₂(B₂)

S

COP=B205 STORE CLOCK

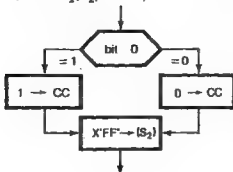
(370)



م الحالية للساعة توضع في كلمة مزدوجة بعنوان St . البتة 31 من ساعة تزداد كل 1,048566 ثانية . ويتم تركيز كود الشرط حسب حالة الساعة .

TS D₂(B₂) S COP=93

TEST AND SET



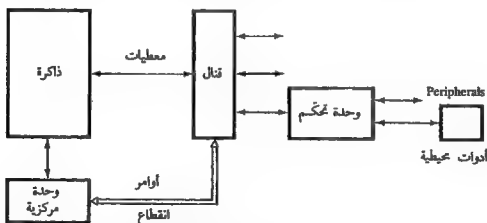
هذه العملية تفحص البتة 0 من البايته بعنوان S₂ وبذلك تضع جميع البتات في 1. يتم تركيز CC. لا يمكن قطع هذه العملية. وتستعمل بشكل خاص للتحكم بتقاسم المصدر بين عمليتين (Processus و CROCUS) systemes des exploitation des ordinateurs, Dunod)

2.19 . الإدخال - الإخراج

سنعرض هنا للعمليات المهمة لإجراء المداخل والمخارج . بإمكان القارئ ، عند القيام باختباراته ، إجراء إدخال - إخراج باستعمال حلقات من فورتران ، مثلاً ، أو بفضل وجود ماكرو تعليقات موجودة على النظام الذي يعمل عليه . سنعود بعد قراءة العموميات إلى دراسة ماكرو تعليقات الإدخال - الإخراج .

1.2.19 . تعريف وأولية الإدخال - الإخراج

عملية الإدخال - الإخراج هي عملية نقل المعطيات من الذاكرة إلى الأدوات المحيطة وبالعكس ويتم بأمر من الوحدة المركزية تحت مراقبة وتنفيذ ألتقال .



عند إطلاق العملية فإنها تدور دون تدخل الوحدة المركزية . يظهر ألتقال وكأنه مُعالج مُستقل وتُخصّص لتبادل المعطيات بين الذاكرة والجهاز المحيطي . وبشكل عام ،

يوضع البرنامج الذي طلب الإدخال / الإخراج في الانتظار حتى إنتهاء عملية الإدخال / الإخراج . وهذا يعني أن تنفيذ معلق خلال مدة الإدخال / الإخراج . وهو يفقد مصادر الوحدة المركزية التي يمكن أن تُخصَّص إلى برامج أخرى مُنتظرة التنفيذ . بعد إنتهاء عملية الإدخال - الإخراج - وهذا ما يتم إعلام النظام به بواسطة الإنقطاع - سيكون بإمكان البرنامج المقطوع أن يُعاود العمل ، وسيوضع في سجل البرامج التي تنتظر مصادر الوحدة المركزية . هنا يدخل موضوع المزامنة المفروض من الإدخال - الإخراج . يتم تأمين هذا التنظيم والإدارة بواسطة برامج (رُجُل) خاصة من نظام التشغيل وهذا هو السبب الذي لأجله لا يستطيع البرنامج أن يُوجَّه بالكامل عمليات الإدخال - الإخراج الخاصة به . فهو يعطي فقط الإشارات اللازمة لنظام التشغيل ليؤمن حسن تشغيل ودوران برنامجه .

2.2.19 . المعلومات الضرورية لعملية إدخال - إخراج
فلنفكّر من خلال مثل من فورتران . لنفترض عملية كتابة على الطابعة I و J هي متحولات صحيحة .

```
WRITE(6,1000) I,J
1000  FORMAT(1X,'I= ',I5,'J= ',I5)
```

إذا كانت قيمة I و J هي على التوالي -4532 و 3 ، نحصل إذاً على :

```
I=Δ-4532ΔJ=ΔΔΔΔΔ3
```

حيث Δ ترمز إلى الفسحة (البياض) الفارغة .

هذه التعليمة في الإدخال - الإخراج المستوحاة من لغة متطورة تغطي مرحلتين مختلفتين .

- لتحويل المتحولات الصحيحة I و J (ثنائي بفاصلة ثابتة) إلى سمات قابلة للطباعة .

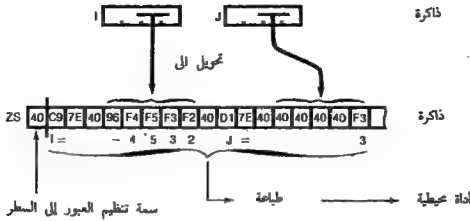
وُسن عملية الإدخال - الإخراج ، أي تبادل المعطيات .

المخطط اللاحق يوجز العمليات .

النسق FORMAT يُمثّل إذاً القناع الذي تكلمنا عنه عند دراسة تعليمات الطباعة . المرحلة 1 تتم تحت تحكم البرنامج ، المرحلة 2 تقع على عاتق القنّال .

نلاحظ إذاً أنه من الضروري معرفة :

- نوع الأداة المحيطة (رقم الوحدة المنطقية ، بلغة فورتران) ،
- العنوان ZS للمنطقة المطلوب طباعتها .



- طول ZS بالبايتات ،
- نوع الأمر (READ أو WRITE) .

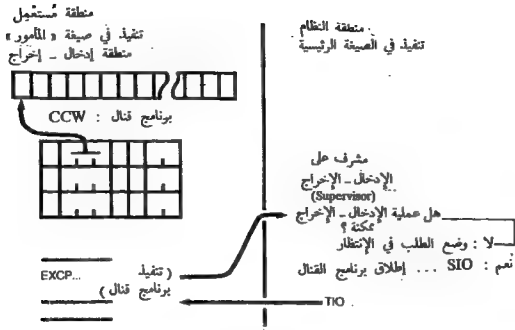
هذه المعلومات إضافة إلى معلومات أخرى ، لأن عمليات الإدخال - الإخراج هي في الواقع أكثر تعقيداً ، يتم وضعها في كلمة مزدوجة للتحكم بالقناة تدعى CCW (Channel command word) : كلمة أمر للقناة .

يلعب القناة دور الحاسب لأنه قابل للبرمجة . ستدعى «برنامج قناة» أو «برنامج وحدة تبادل» ، مجموعة الكلمات CCW المكونة من أوامر متتالية تتحكم بالمحيط .

الأدوات المحيطية هي عبارة عن مصادر قابلة للتقسيم والتوزيع بين عدة مستعملين . يصبح إذاً من الضروري معالجة النزاعات التي قد تولد من جراء طلبات متزامنة لنفس المصدر . لهذا السبب فإن مسؤولية إطلاق برنامج القناة تقع على عاتق نظام التشغيل الذي سيتحقق من توفر القناة والوحدة المحيطية . وبشكل آخر ، بإمكانه أن يأخذ بعض القرارات في حالة حدوث تنفيذ خاطئ لعملية الإدخال - الإخراج . الكلمة - المزدوجة ذات العنوان 40 ، بالنظام السادس عشري ، والتي تدعى CSW (Channel status word) ، تعطي بعض المعلومات حول دوران ومحاولة إطلاق الإدخال - الإخراج . المخطط الوارد على الصفحة التالية يقوم ببعض عمليات الربط بين مختلف العناصر الضرورية للإدخال - الإخراج .

3.2.19 . إدخال - إخراج في المستوى المنطقي

إن تنفيذ عملية إدخال - إخراج بالمستوى الفيزيائي هو أمر معقد . كتابة CCW تتطلب معرفة واضحة بالمحيطات التي تعمل عليها . ونعرف أنه في أغلب الوقت تكون



عمليات الإدخال - الإخراج على المحيطات البطيئة مؤجلة . عندما يقوم المستعمل بتعريف سجل طباعة (حالة 6, WRITE بلغة فورتران) ، فإن هذا السجل هو أولاً مكتوب على قرص مغناطيسي وبعد ذلك ، بواسطة برنامج خاص ، يُؤخذ لإجراء طباعة نهائية . وفي المجموع فإن رقم الوحدة المنطقي ، يُناسب أولاً فيزيائياً سجل قرص مغناطيسي وبعد ذلك سجل الطباعة . هذه العملية ، التي تحاول تبسيط إدارة المصادر المركزية والمحيطية ، تؤدي إلى زيادة الصعوبة في تنفيذ عملية الإدخال - الإخراج الفيزيائية . من جهة أخرى ، فإن تنظيم عملية إدخال - إخراج يؤدي إلى درء (Bufferization) لمناطق إدخال - إخراج . نعرف أيضاً أنه يوجد عدة تنظيمات نموذجية للسجلات وعدة طرق للبلوغ . هذه الشروط تفرض على المستعمل بأن يأمن بالكامل لنظام إدارة عمليات الإدخال - الإخراج . للقيام بذلك يجب عليه وصف التغيرات الوسيطة المقيدة بواسطة توجيه من نوع DCB (DATA CONTROL BLOCK) . وهو سيوكل عملية الإدخال - الإخراج الخاصة به للنظام بواسطة ماكرو تعليمية خاصة (PUT GET ...) حسب نوع تنظيم السجل الخاص به . هذه الأخيرة هي موضحة في الوثائق OS/VSS2 MVS (DATA Management Macro Instructions) . يقوم النظام بتوليد الكلمات CCW لنفسه ونداء للمشرف الضروري . العملية الأولى للإدخال - الإخراج ستكون مسبوقة بفتح للسجل (ماكرو OPEN) والأخيرة ستكون متبوعة بإغلاق للسجل (ماكرو CLOSE) يسمح بتفريغ الدارء (Buffer) الأخير . المثال التالي يوضح ، بإشراف النظام OS ، عملية قراءة بطاقة مثقوبة وكتابة على الطباعة .

```

OPEN      (CARTE,(INPUT))
OPEN      (IMP,(OUTPUT))
-----
GET       CARTE,ZENTREE
-----
PUT       IMP,ZSORTIE
-----
CLOSE     CARTE
CLOSE     IMP
-----
CARTE     DCB      DDNAME=ENTREE,DSORG=PS,LRECL=80,BLKSIZE=400,MACRF=(GM),
              RECFM=FB,EODAD=SUITE
IMP       DCB      DDNAME=SORTIE,DSORG=PS,LRECL=133,BLKSIZE=665,MACRF=(PM),
              RECFM=FBA
ZENTREE   DS       CL'80'
ZSORTIE   OC       133C'
-----

```

20 . الأوامر المتعلقة بالبنية

وتركيبة البرنامج

منقوم بجميع الأوامر (التوجيهات) المستعملة عند بداية ونهاية البرنامج ، التي تسمح بإعداد عدد المواقع ، وتعريف المرافف القاعدية أو تغيير وتقطيع البرامج .

1.20 . تعريف وشحن مرافف القاعدة
لقد عرفنا العنوان القاعدية (فقرة 2.3) وعرضنا مثلاً على تأويل تعليمية من هذا النوع (فقرة 3.3.6) من الضروري العودة الآن بشكل أكثر تنظيمياً لهذه المسألة :
إهتمامات المبرمج الأولى هي :

- 1 - تحديد واحد أو عدة عناوين قاعدية .
 - 2 - حجز واحد أو عدة مرافف سيتم استعمالها كمرافف قاعدية .
 - 3 - شحن هذه المرافف بالعناوين المناسبة .
- النقطتان الأوليان تتعلقان بمرحلة التأويل ، والنقطة الثالثة تتعلق بمرحلة التنفيذ ولا يمكن أن نُحلّ بشكل نهائي عند التأويل لأن العنوان الفعلي لحزن البرنامج في الذاكرة لن يكون معروفاً إلا في لحظة الشحن .

أ - USING

هو الأمر الذي يسمح للمؤول بتحديد مرافف القاعدة وحساب الإزاحة المطلوبة لعنوان محدد رمزياً (قاعدة ضمنية ، فقرة 2.9) . وشكله هو التالي :

USING Ad. base, numero des registres de base

رقم مرصف القاعدة وعنوان قاعدة USING

«Ad. Base» هو تعبير مطلق أو قابل للنقل يعتبره المؤول عنواناً قاعدياً . هذا الأمر لا يؤلّد أية تعليمية ولذلك فهو لا يزيد من قيمة عداد المواقع . وهو ينتهي من البرنامج المؤول .

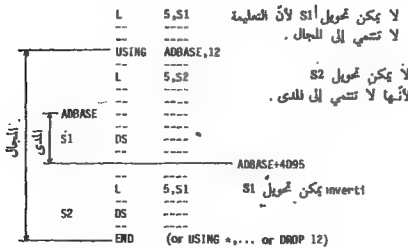
مثلاً :

- (1) USING ADBASE,12
- (2) USING ADBASE,12,11,10
- (3) USING *,15

الإزاحة هي كمية مكوّنة من 12 بتة لا تزيد عن 4095 . وبالتالي ، فإن مدى مصرف القاعدة 12 ميمتد من ADBASE إلى ADBASE + 4095 . عندما يزيد البرنامج عن 4096 بايتة يجب إستعمال الشكل (2) أو عدة أوامر USING لتحقيق العنوان . في الشكل (2) يفترض المؤول أنّ للمصرف 12 يحتوي على القيمة ADBASE ، والمصرف 11 القيمة ADBASE+4096 والمصرف 10 القيمة ADBASE + 8192 . في الشكل (3) يفترض المؤول إن العنوان القاعدي هو القيمة الحالية لعداد المواقع .

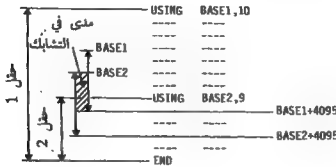
قواعد الإستعمال

لنميز « مدى » المصرف القاعدي من الحقل المغطى بواسطة تعليمة USING .
مدى للمصرف القاعدي لا يتعلّق سوى بالعنوان القاعدي المذكور في الأمر وليس بموقع USING . ويمتد من ADBASE إلى ADBASE+4095 . هذا يعني إن جميع الرموز التي تنتمي إلى المنطقة يمكن أن تعنون بناء على انتهاء التعليقات التي تُرجع إليها إلى « الحقل » .
الحقل USING يمتد من الأمر (التوجيه) USING حتى نهاية (END) الزوجلة .
الأمر الآخر USING يُحدّد نفس المصرف أو يضع الأمر DROP النهاية للحقل السابق .
المثل التالي يوضح ذلك .



حالة استعمال عدة أوامر USING

عندما يتشابك مدى عدة مرافف ، فإن المؤول يحدد بشكل جلي العناوين الرمزية المشتركة لكلا المدينين باختيار عنوان قاعدي ذلك الذي ينتج أصغر إزاحة . إذا كانت العناوين القاعدية متشابهة (BASE1 و BASE2 هي ذاتها) ، فهو يختار رقم المرصف الأكبر . إذا كانت العناوين مختلفة ولكن المرافف متشابهة فإن الأمر الثاني USING يقطع مدى الأول



ب- شحن مرافف القاعدة

يتوجّه الأمر USING إلى مرحلة التأويل (assembling) . يجب على المبرمج أن يتوقع تعليمة تقوم ، عند التنفيذ ، بتخزين المرافف القاعدية بالعناوين الفعلية الضرورية . هذه العناوين لا يمكن أن تكون معروفة في لحظة التأويل (assembling) لأنها تتعلق بنقطة الشحن (فقرة 4.6) . المشكلة هي إذاً في كيفية معرفة طريقة استرجاع هذه العناوين . نستعمل لذلك تقنيتين : الطريقة الأولى تستعمل حالة خاصة في استعمال BALR : حيث R_2 هو المرصف 0 (فقرة 4.12) . هكذا فمن الممكن كتابة :

```
BALR 12,0
USING *,12
```

يُخزّن عنوان التعليمة BALR زائد 2 (طول التعليمة) في المرصف 12 وهذا العنوان (●) يُحلّد كقاعدة .

الطريقة الأخرى تقوم على استعمال إتفاق عادي من النظام OS (فقرة 5.21) بحججه يُخزّن النظام في المرصف 15 عنوان نقطة الدخول إلى البرنامج الذي ينتقل التحكم إليه . هذه هي طريقتنا المفضلة . سنختار كعنوان قاعدي عنوان بداية (نقطة الدخول) إلى البرنامج .

DEBUT CSECT

USING DEBUT,12
LR 12,15

وبالتالي ، وحدها التعليقات التي لا تستعمل عناوين رمزية يمكن أن تظهر قبل شحن المرصيف القاعدي .

ج- DROP

التوجيه أو الأمر R_1, R_2, \dots, R_n DROP يُشير إلى المؤول لكي لا يستعمل المراصف R_1, R_2, \dots, R_n كمراصف قاعدية .

2.20 . تقطيع البرامج

كل برنامج مهم يجب أن يكون مقطّعا ، أي مقسماً إلى قطع (زجل module) مستقلة . هذا ما يؤمن لنا بعض الاهتمامات : تبسيط البرامج وتقيص طول المهام ، إعطاء البرنامج كاملا تركيبة زجلية تسمح بتسهيل عملية تعديل البرنامج ، تسهيل عمل الفريق (العمل الجماعي) ... ونحصل على ذلك بتقسيم البرنامج إلى عدة أقسام - مصدر ، باستعمال الإمكانات التي تضعها البرامج الثانوية بتصرفنا (أنظر الفصل 21) ، وباستعمال أوامر (توجيهات) التقسيم .

قسم مهم من عمل المؤول يقوم على ربط الرموز الموجودة في الزجل (الأقسام) بعناوين محددة على شكل قاعدة ، مؤشر وإزاحة . ينتهي المؤول من العمل عندما يلتقي الأمر END الذي يشير إلى نهاية الزجلة . تتألف الزجلة المصدر من مجموعة من التعليقات المؤولة في مرة واحدة .

1.2.20 . رموز داخلية ، رموز خارجية

يمكن تصنيف الرموز التي يلتقيها المؤول في زجلة مصدرية ، في عدة طبقات .

1- الرموز المطلقة .

2- الرموز المنقولة التي تظهر في منطقة الـوسم . وهي تسمح عادة ببلوغ تعليمة أو معطى ما . ولا يمكنها أن تظهر إلا مرة واحدة في منطقة الـوسم خوفاً من التعريف المزدوج . كما أنها داخلية ضمن زجلة المنبع ويقوم المؤول بربطها بعنوان على شكل قاعدة وإزاحة . ويقوم بتخزينها في جدول الرموز للنقولة (المترجمة) .

3- الرموز التي تظهر في منطقة الـوسم ولكن من النوع « نقاط الدخول » . وتنتمي إلى زجلة المصدر ولكنها قد تكون قابلة للتسمية بواسطة أسماء من خارج هذه الزجلة . من الممكن تصنيفها في طبقتين : طبقة الرموز المستعملة . في تسمية التعليقات ، وطبقة تلك التي تستعمل لتسمية مناطق المعطيات . يقوم المؤول بتخزينها في جدول

الرموز الخارجية ESD (External Symbol Dictionary) حتى لو كانت داخلية في زجلة المصدر . رمز واحد على الأقل ينتمي إلى الفئة الأولى : الرمز الذي يشير إلى التعليمة الأولى للتنفيذ . إذا كان هذا الأمر غائباً فإن المؤول يختار كتقطعة دخول عنوان التعليمة الأولى من البرنامج ويُخزّنه في ESD . يجب تعداد الرموز من النوع نقاط الدخول في الأمر ... ENTRY SYMB1, SYMB2, إذا لم تكن معتبرة كتقاط دخول إذا كانت مستعملة لتسمية القطعة (الزجلة) .

4 - الرموز التي تظهر في زجلة منطقة العوامل ولكن غير الموجودة في منطقة الوسم . هذه الرموز تنتمي إلى زجل مصدرية أخرى ولا يستطيع المؤول أن يربط عنواناً بها ؛ وهو يعهد بهذه المهمة إلى مُنقّح الأربطة (link editor) أو إلى الشاحن ، وذلك بتخزينها في ESD . تُعتبر هذه الرموز خارجية بالنسبة لزجلة المصدر . إنَّها عبارة عن نقاط دخول إلى زجل أخرى وإذا فهي تنتمي إلى إحدى الطبقتين المذكورتين في 3 . ويجب أن يكون مصرحاً عنها وكأنها خارجية بواسطة الأمر EXTRN SYMB1, SYMB2 ... إلا إذا كانت عبارة عن أسماء برامج ثانوية مصرحاً عنها في ثابتة بعنوان من النوع V .

2.2.20 . أوامر التقسيم

هذه الأوامر تشير إلى بداية أو نهاية قسم من زجلة المصدر .

[تعبير منقول (مترجم)] END

يشير إلى نهاية زجلة المصدر . العنوان المناسب للتعبير المنقول يُخزّن في ESD . إنَّه بشكل عام عنوان أول تعليمة للتنفيذ .

```

CSECT
-----
ALPHA
-----
END      ALPHA

```

يُعرف ALPHA كتقطعة دخول إلى البرنامج .

قسم التحكم (Control section) هو عبارة عن قطعة منقولة من البرنامج (قابلة للترجمة) . هذا يعني بأنّه يجب أن تربط بها مرصيف قاعدة واحداً على الأقل ، مما يجعل هذه الوحدة قابلة للنقل والترجمة بشكل مستقل عن باقي البرامج . وهي تبدأ بحدود كلمة مزدوجة . يمتد قسم التحكم من بداية القسم حتى إلتقاء قسم آخر .

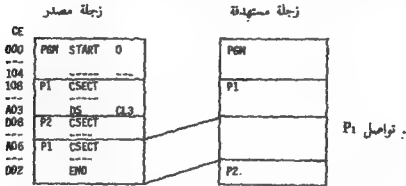
[symbole] START [constante]

[ثابتة] START [رمز]

يقوم بإعداد قسم التحكم الأول بزجلة المصدر . الثابتة الاختيارية تسمح بإعطائه قيمة أولية إلى عداد المواقع . يُخزّن الرمز في ESD .

[Symbol] CSECT

يعرّف عن قسم التحكم أو يُؤشر إلى قسم داخلي . الإلقاء الأول للرمز يشير إلى بداية القسم ، والإلقاء التالي لنفس الرمز يُشير إلى مواصلة القسم . يعمل المؤرّل قسماً بعد قسم : يختلف قطع القسم تكون موجودة متصلة في نفس الزجلة المستهدفة (object module) ، هكذا في المثل التالي ، يتمّ تأويل تواصل P1 قبل P2 . من هنا نحصل على قاعدة كيفية تطوّر CE .



تُخزّن الرموز PGM ، P1 و P2 في الجداول ESD . وهي تُمثّل نقاط الدخول . تشير إلى أن جميع أقسام التحكم يجب أن تُعرّف بواسطة رمز ما عدا واحداً . يمكن أن يُعرّف بواسطة اسم أبيض . يجب على كل قسم ، وهذا موجود في التعريف ، أن يتمتع بمرفص قاعدة . ويُعرّف المؤرّل العناوين الفيزيائية للقسم باستعمال هذا المرفص القاعدي الذي يجب أن يُشعّن مع قيمة العنوان المناسب . يمكن لقسم التحكم أن يبدأ على الشكل التالي :

[symbol] CSECT (أو START للأولى)
BALR RBASE,0 (حيث RBASE هو المرفص القاعدي)
USING *,RBASE

ستعرض عليكم حلاً آخر لشحن المرفص القاعدي في الفصل 21 .

القسم الوهمي (dummy section) هو عبارة عن قسم مستعمل فقط لوصف المعطيات دون حجز لمواقع لها في الذاكرة ويسمح إذا بتعريف رموز دون ربطها بعناوين في لحظة كتابة القسم الوهمي . المثل التالي سيوضح ذلك :

لنفترض البرنامج التالي الذي يستعمل المنطقتين Z1 و Z2 المنفصلتين فيزيائياً مع أنها بتركيبة متشابهة . ستقوم بتعريف التركيبة المشتركة في تركيبة وهمية تدعى ENREG وسنطبقها على Z1 و Z2 عندما يصبح ذلك ضرورياً .

Z1	OS	CL80	
Z2	OS	CL80	حجز المناطق
	---	---	
	USING	ENREG,4	تعريف العنونة بالنسبة للقسم الوهمي
	L	4,=A(Z1)	
	---	---	
			تطبيق تركيبة القسم الوهمي على Z1
	L	4,=A(Z2)	
	---	---	
			تطبيق تركيبة القسم الوهمي على Z2
ENREG	DSECT		
NUMERO	OS	CL4	
MONTANT	OS	CL10	
NOM	OS	CL20	
ADRESSE	OS	CL46	

[symbole] DSECT

يُعرف عن بداية أو تواصل القسم الوهمي . عنونة القسم يمكن أن تتم بفضل وجود الرمز الموجود قبل DSECT أو بفضل وجود أي رمز في الوصف . يُوضع عدّاد الرموز دائماً في صفر عند بداية DSECT . يُخزّن الرمز في ESD . من هنا نلاحظ البساطة الناتجة عن هذا المفهوم . والبرمجة ستكون مُبسّطة ومن هنا يتجّ إقتصاد في استعمال الرموز .

القسم المشترك يسمح لعدة زجل مصدر ، مؤولة بشكل منفصل ولكن متّحلة فيها بينها بواسطة منقح الأربطة ، أن تتقاسم نفس منطقة التنفيذ . سنستعمل هذه المنطقة :

- لإيصال المعطيات بين زجل المصدر (فورتران ومؤول مثلاً) ،
 - كمنطقة عمل مؤقتة لإحدى الزجل بشرط ألا تستعمل في نفس الوقت .
- عند المعالجة بالمؤول سيتم حجز موقع لكل زجلة ، ولكن عند المعالجة بواسطة مُنقّح الأربطة فإن المناطق المشتركة مستحد ، فقط ستحتفظ المنطقة ذات الحجم الأكبر .

[Symbol] COM [رمز]

تُعرف عن منطقة مشتركة . يسمح النظام OS بوسم المناطق ولكن النظام DOS لا يسمح بذلك (لا يوجد رموز) . من الضروري ، في كل زجلة مصدر ، أن يتم

إجراء عنونة بشكل شبيه بما جرى في DSECT . يوضع عداد المراكز في صفر عند بداية القسم .

3.2.20 . مُنقَح الأربطة (link edition)

الفقرات السابقة تسمح لنا بفهم وبشكل أفضل عمل مُنقَح الأربطة والشاحن (loader) .

مع الزجلة المستهدفة ، يقدم المؤول إلى مُنقَح الأربطة جدولاً ESD لكل زجلة مصدر . نجد في الجدول ESD أسماء الرموز من الفئتين 3 و4 (فقرة 2.2.20) . في كل رمز نجد كود العملية من نوع الأمر للربط بها . إذا كان الرمز من نوع نقطة الدخول ، فإن عنوانه هو في الزجلة المشار إليها . بالنسبة للزجلة المصدر المذكورة في الفقرة 4.2.20 ، فإن الجدول ESD يكون على الشكل التالي :

EXTERNAL SYMBOL DICTIONARY

SYMBOL	TYPE	ID	ADDR	LENGTH	LDID
PC	PC	0001	000000	00001C	
ALPHA	ER	0002			
PI	SD	0003	000020	00000C	
DEBUT	ER	0004			
SP	ER	0005			

يكون نوع الرمز على الشكل التالي :

كود	مناسب للأمر
PC	بدء رسم CSECT أو START
SD	مع رسم CSECT أو START
DN	DDNN
XD	DXD أو DSECT خارجي (1)
LD	ENTRY
ER	EXTRN DC V(....) أو ثابتة بعنوان (2)
WX	WXTRN (2)

في مقابل هذه المعلومات المرتبطة بكل زجلة ، فإن مُنقَح الأربطة يقوم بالإجابة على الطلبات الخارجية ، أي يقوم بإجراء التناسب بين الأسماء الموجودة في مختلف ESD . وإذا لم يكن بإمكان المُنقَح أن يحل مشكلة الطلبات الخارجية بسبب جدول الزجل ESD المطلوب ربطها ، فهو يقوم بعملية بحث متتظمة في المكتبات التي يقدر على بلوغها .

(1) CXD ، DXD ، DSECT الخارجية هي غير مشروحة في هذا الكتاب .
(2) WXTRN تقوم بحل نفس الدور الخاص بـ EXTRN . في ما يتعلق بالساح مُنقَح الأربطة بالبحث الأتوماتيكي عن الرموز بداخل المكتبة ، فإن WXTRN تمنع هذا البحث .

4.2.20 . الشحن (loading)

يقوم الشحن على تخزين البرنامج في الذاكرة بدءاً من عنوان مُحدد . كما رأينا في الفقرة 2.3 ، العناوين المتقولة لا يجب أن تتعلل خلال هذه العملية . والأمر ليس كذلك بالنسبة لثوابت العنوان . يقوم الشاحن بخزن العناوين الفعلية للمتأثرات المطلوبة في الذاكرة .

يجب على المؤول أن يرسل إلى الشاحن مواقع المناطق المطلوب إعادة حسابها . يستعمل لهذا الهدف RLD (Relocation Dictionary) حيث تتواجد عناوين ثوابت العنوان . الجدول ESD في الملأ أعلاه هو موجود في الفقرة 3.2.20 . نذكر بأن DC V (SYMB) يعادل :

EXTRN SYMB

DC A(SYMB)

يُحتفظ باستعمال ثوابت العنوان من النوع V للتعريف عن عنوان تفريع (إسم قسم ، إسم برنامج ثانوي . . .) الرمز SYMB يُخزن في ESD . ويقوم المؤول بتصفير الثابتة .

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
000000				1	START	0
			00000	2	USING	*+12
				3	EXTRN	ALPHA
000000	5B30 C010	00010		4	DEBUT	3=A(ALPHA)
000004	5B50 C014	00014		5	L	S=A(BETA)
000008	FFFFFFFF			6	ZONE	R*-1
000020				7	DC	CSECT
				8	P1	USING
000020	5B50 C01B	0001B	00020	9	BETA	S*=V(SP)
000024	00000000			10	ADR	A(DEBUT)
000028	00000000			11	DC	V(DEBUT)
				12		END
000010	00000000			13		A(ALPHA)
000014	00000020			14		A(BETA)
000018	00000000			15		V(SP)

RELOCATION DICTIONARY

POS-ID	REL-ID	PLAGS	ADDRESS
0001	0002	0C	000010
0001	0003	0C	000014
0001	0005	1C	000018
0003	0001	0C	000024
0003	0004	1C	000028

سنفحص في المخطط التالي كيفية تطوّر القيمة المأخوذة من قبل ثابتة عنوان من التأويل إلى الشحن :

نستطيع بلوغ SYMB2 بامتثال ثابتة العنوان A(SYMB2) التي يقوم الشاحن بإعدادها بشكل مناسب . وفي المقابل ، فإن التعليمة 16 SYMB1 3, L يمكن أن تكون مؤولة .

2- عند التنفيذ ، تكون المشكلة مختلفة : التعليمة SYMB1 3, L هل ستسمح بالبلوغ إلى SYMB1 ؟

قد يسمح لنا التأويل المناسب للتعليمة بهذا الافتراض . هكذا فعملياً هذه التعليمة تسمح عند التنفيذ ، ببلوغ SYMB1 بشرط أن تكون القاعدة 12 المُنونة SYMB1 تحتوي على العنوان P1 المناسب . ولكن لا شيء مؤكداً ، في مثل معاكس ، يكفي أن يكون القسم P2 مُنفذاً قبل القسم P1 كي لا تكون القاعدة 12 مشحونة بشكل مناسب . إضافة لذلك ، فإن أي مراجعة من هذه الطبيعة تناقض تعريف قسم التحكم . وبالتالي فإننا سنراجع SYMB1 في P2 بفضل وجود ثابتة العنوان .

يظهر إذاً ويوضح أن الأقسام يجب أن تُعتبر كوحدات مُستقلة في نفس الوقت الذي تكون فيه الزجل المصدريّة منفصلة عند التأويل . الاتصال الرمزي بين الأقسام سيتم دائماً بواسطة ثوابت العنوان . هذه التقنية تسمح بتفادي العقبة المثارة أعلاه وتسمح ببلون مشكلة بتوزيع الأقسام في مختلف زجل المصدر .
وإيجاز ، فإن تفرّيع القسم سيتم بواسطة :

$$\begin{array}{l} L \quad R_i = V(P1) \quad (= A(P1)) \\ BR \quad R \end{array} \quad \text{حيث } R \text{ هو مصرف عام}$$

R هو مصرف عام ، بشكل عام المصرف 15 حسب إتفاقات الربط المعروضة في الفقرة 4.21 .

بلوغ الرمز يتم بواسطة :

$$\begin{array}{l} L \quad R_i = A(SYMB) \\ L \quad R, O(R) \end{array}$$

6.2.20 . ختام حول التقسيم

يعطي التقسيم وسيلة لتجزئة زجلة المصدر إلى زجل مُستقلة . عند إجراء التقسيم فإن كل شيء يجري كما لو كانت الزجل المصدريّة مترابطة .

نحرص على عدم بلوغ ، في نفس القسم ، رموز لا تنتمي إلى هذا القسم . وإذا كنا نرغب ببلوغ رموز خارجية فسنستعمل الطريقة المعروضة في الفقرة 4.2.20 ، تاوكين إلى الشاحن مهمة إجراء الوصلة بواسطة ثوابت العنوان .

3.20 . الأوامر التي تُغيّر عدّاد المواقع
ORG عبارة عن تعبير منقول أو مطلق . هذا الأمر يؤدي إلى تغيير الازدياد الطبيعي لعدّاد المواقع . وهو يسمح بشكل خاص بإجراء إعادة تعريف أو حجز مكان من الذاكرة . إذا كانت منطقة العناصر (القياسات) فارغة ، فإن ORG يعطي عداد المواقع CE القيمة التي كانت موجودة فيه عند آخر تعديل بواسطة ORG . لا يمكن أن يكون القياس (argument) مبلوغاً في البداية .

قيمة العداد
CE

000	TABLE	DC	XL256*40*
00A		ORG	TABLE+10

000		ORG	---

LTORG عبارة عن أمر بدون قياسات . وهو يشير إلى المكان الذي يجب أن تُؤوّل فيه الثوابت الحرفية . في غياب هذا الأمر فإن تأويلها سيتم في نهاية أول قسم .
CNOP b, w يؤدي ، بحكم عدم إجراء أية عملية ، إلى زيادة قيمة عداد المواقع إلى الحد الأقرب لنصف كلمة ، كلمة أو كلمة مزدوجة حسب قيمتين b و w .

CNOP	0,4	بداية كلمة
CNOP	2,4	وسط كلمة
CNOP	0,8	بداية كلمة مزدوجة
CNOP	2,8	النصف كلمة الثاني من كلمة مزدوجة
CNOP	4,8	النصف كلمة الثالث من كلمة مزدوجة
CNOP	6,8	النصف كلمة الرابع من كلمة مزدوجة

4.20 . أوامر التحكم باللوائح
ICTL يسمح بتعديل الإطار النموذجي (الأعمدة 1 ، 16 و 71) للتعليقات .
ISEQ يسمح بالتحقق من الترتيب المتتالي للبطاقات .
COPY يسمح بنسخ قسم من النص المصدر في المكتبة .
EJECT يؤدي إلى ظهور التعليمة التالية في رأس الصفحة التالية من اللائحة .
وهو مفيد لتوضيح نص البرنامج .
SPACE n يسمح بإدخال عدد n من الأسطر الفارغة في اللائحة .

PRINT [ON, GEN, NODATA
OF, NOGEN, DATA]

يسمح بالمحافظة على أو بإلغاء اللائحة (Listing) ، توليد الماكرو وتعليقات توليد المعطيات .

«سلسلة» TITLE يسمح بطباعة عنوان من 100 سمة في رأس كل صفحة .
PUNCH, REPRO يسمحان بتقييد البطاقات .

5.20 . أوامر مُستعملة بإشراف النظام OS فقط
OPSYN يسمح بتعريف مجموعة كود العمليات الخاصة المُراددة للأكواد IBM .
هذا الأمر يمكن أن يكون مفيداً بشكل خاص لاستبدال كود - عملية خاص بماكرو
عملية .

من الممكن إذاً تبديل الكود الحرفي BE ، BNE ، . . . للماكرو حيث الأسماء
سيُصرَّح عنها بشكل مرادف بسبب وجود OPSYN . هذه الماكرو تعليمات تولّد كلمة
تُخزَّن فيها نتيجة الاختبار الذي يسبق تعليمة التفرع بالشكل V أو F أو O أو N « وبعد
ذلك تقوم بالتفرع المناسب باستعمال التعليمة BC أو BCR . هذه السمات V أو F
ستكون مرئية في العملية DUMP (دلق) وتسمح بمتابعة أثر تنفيذ البرنامج (Trace) .
بالإمكان تمييز مختلف الأسماء المُولدة بواسطة SYSNDX & (فقرة 7.2.22) .

بعد مرحلة الإطلاق في العمل ، فإن إلغاء الأوامر (التوجيهات) OPSYN يؤدي
إلى تفادي إدخال ماكرو التعليمات والبدء بتنفيذها .

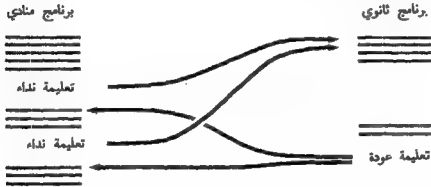
من الممكن أيضاً استعمال هذا الأمر لجعل بعض التعليمات غير عملية وذلك
بجعلها مرادفة للتعليمة NOP (لا عملية) .

PUSH و POP . من الممكن عند كتابة البرنامج أن نقوم بشحن مرصف القاعدة
بسرعة وأن نستعيد القاعدة القديمة لاحقاً . هذا يمكن أن يتم مثلاً ، عندما تستعمل
إحدى ماكرو التعليمات قاعدة شخصية . بعد التبديل ، بواسطة المؤول ، يجري فقدان
القاعدة القديمة . يسمح الأمر PUSH بتخزين المراسف وعنوان القاعدة وصيغ الأمر
PRINT داخل مكدس (Stack) ⁽¹⁾ . POP يُعاود إسترجاع المفهوم القديم بواسطة
إستخراج لآخر كلمة مكدسة .

(1) المكدس هو عبارة عن جدول مُنظَّم حسب التقنية «الدخل أخيراً هو الخارج أولاً» .

21 البرامج الثانوية

البرنامج الثانوي هو عبارة عن سلسلة من التعليمات التي يتم تنفيذها بطلب من تعليمية نداء (Call) . عندما ينتهي تنفيذ البرنامج الثانوي يعود العمل بالبرنامج المُنَادِي وبالتعليمية التي تتبع مباشرة تعليمية النداء . المخطط التالي يوضح هذه الأولوية :



كل شيء يجري كما لو كانت تعليمات البرنامج الثانوي داخلية في مكان تعليمية النداء .

بإمكاننا تقسيم البرنامج الى مهام (task) ، كل مهمة يتم حلها بواسطة برنامج ثانوي . إعداد البرنامج يكامله يصبح سهلاً ، والأقسام تصبح صغيرة . هذه الأولوية تسمح بتفادي إعادة كتابة التعليمات المتشابهة عندما يجب تنفيذ البرنامج في مختلف مستويات البرنامج المُنَادِي . وتطرح هذه التقنية مشكلتين :

- تخزين عنوان العودة (العنوان الذي يتبع مباشرة عنوان تعليمية المُنَاداة) ،
- إنتقال المتغيرات الوسيطة .

مشكلة إنتقال المتغيرات جرت إثارتها في إطار تقسيم البرنامج ولكن البرنامج الثانوي لا يُشكّل بالضرورة قسم تحكم

1.21 - البرنامج الثانوي وقسم التحكم

التقسيم هو عبارة عن عملية تتعلق بالتأويل ، تنقيح الأريطة والشحن : أما مفهوم البرنامج الثانوي فلا يتعلق سوى بالتنفيذ . مناداة البرنامج الثانوي تؤدي ، عند التنفيذ ، إلى تعديل الدوران المتتالي للتعليمات .

هكذا ، فلا شيء يعترض بأن يكون البرنامج والبرنامج الثانوي تابعين لنفس القسم . ولكن هذا النوع من التنظيم لا يُقدّم جميع الفوائد التي نتظرها من البرنامج الثانوي . فهو يربط البرنامج بالبرنامج الثانوي بينما نرغب نحن بجعل البرنامج الثانوي قابلاً للطلب والدعوة من جميع الأقسام أو الزجل . وهو لا يشكل تحسّناً باتجاه تركيبة زجلية . وبالتالي لا يستعمل إلا عندما يكون البرنامج الثانوي مرتبطاً بشكل كبير منطقياً بالبرنامج المنادي .

في أغلب الأحيان يُفضّل إستعمال إمكانيات التقسيم : حيث يشكل البرنامج الثانوي قسماً من البرنامج . من المحتمل ، منذ لحظة تصوّر البرنامج الثانوي ، إستعمال هذه الزجلة في مُعالجات أخرى . يُفضل معالجة مشكلة الاتصال بين البرنامج / البرنامج الثانوي كوصلة برنامج خارجي تسمح بإمكانية تفكيك عمليات التأويل دون تعديل في الأقسام .

2.21 . تفريع إلى برنامج ثانوي والعودة

مناداة البرنامج الثانوي ليست سوى قطع إلزامي للدوران المتتالي للتعليمات ولكن مع تخزين للعنوان التالي الذي يتبع تعليمة المائدة بشكل يسمح بمعاودة العمل بالبرنامج المقطوع . تتمتع كل مكنة بأولية خاصة للتفريع مع عودة . يستعمل النظام 360/370 التعليمتين BAL و BALR اللتين رأيناهما في الفصل 12 .

BAL R1,D2(X2,B2)
BALR R1,R2

يكون عنوان العودة مخزناً في المرصف R1 . يكفي إذاً في نهاية البرنامج الثانوي أن نشحن عدّاد البرنامج بالقيمة المخزّنة في R1 بواسطة التعليمة BCR 15,R1 مثلاً . نحصل إذاً على التركيبة التالية :

البرنامج المنادي

البرنامج الثانوي SP

```

-----
-----
-----
L      R2,=A(SP) ( V(SP) أو )
BALR   R1,R2  إذا كان SP خارجياً
-----

```

(تخزين المرصفتين وتعريف القاعدة)

(إعادة مضمون المراضف إلى الذاكرة)
BCR 15,R1

إذا كانت BALR موجودة على العنوان ALPHA ، فإن BCR.15,R1 تعيد تخزين ALPHA +2 في عداد البرنامج (CO) .
كان بإمكاننا إستعمال BAL بأحد الأشكال التالية :

- 1°) BAL R1,SP إذا كان SP عبارة عن مرجع داخلي
- 2°) L R2,=A(SP) ou =V(SP)
BAL R1,DEPLAC(R2)

الشكل الذي يسمح ، بواسطة حساب بسيط لـ DEPLAC ، بالحصول على مداخل متعلّقة في SP .

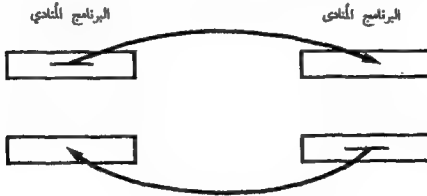
نلاحظ أنه لا يوجد فرق أساسي بين التفرعات إلى برامج ثانوية خارجية أو داخلية . وحده تعريف ثابتة العنوان الخارجي هو إلزامي في الحالة الأولى .

3.21 . إنتقال المتغيرات الوسيطة

المشكلة الثانية في عملية الاتصال بين البرنامج والبرنامج الثانوي تكمن في عملية تبادل المعطيات . إن تقنيات عبور المتغيرات هي متعلّقة ويمكن للقارئ أن يتصور الطريقة الأفضل لمسألة . ولكن من المفيد هنا أن نعرض الطرق العامة التي تساعد على الاختيار . تُستعمل اللغات المتطورة بطريقتين أساسيتين : لانتقال المتغيرات مباشرة بالقيم والانتقال بال عناوين .

إنتقال المتغيرات حسب القيم

ويكمن في نسخ القيمة المطلوب إزsalها إلى منطقة معروفة من البرنامج المتأدى .



هذه المنطقة يمكن أن تكون خلية في الذاكرة مركزية (Local) في البرنامج المُنَادى أو مرصفاً. تستعمل هذه التقنية ، مثلاً في لغة فورتران ، لاعادة قيمة إحدى الدوال إلى البرنامج المُنَادى . وبشكل عام فإن النتيجة تُخزَّن في المرصف 0 بواسطة البرنامج المُنَادى .

نلاحظ إنه إذا كانت B عبارة عن متحوّلة مركزية من البرنامج المُنَادى ، فإن أي تعديل في B لا يؤدي إلى أي تغيير في الخلية A .

وفي لغة المُول ، يمكن أن تُحل مشكلة التبادل بالقيم بواسطة النقل بالمراسف ، حيث يحدد المبرمج طريقة لاستعمال المراسف .

البرنامج المُنَادى

```

-----
L      1,A
L      2,B
SP      3u SP
تفريع إلى SP

```

البرنامج المُنَادى

```

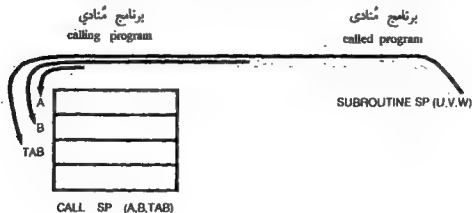
-----
ST      1,U
ST      2,V
-----

```

نشير إلى أن هذه الأولوية هي غير متوافقة مع تبادل الجداول . فعندئذٍ تتطلب مكاناً كبيراً من الذاكرة . هذه الطريقة هي غير مناسبة إلا عندما يكون عدد المعطيات المطلوب إرسالها قليلاً .

انتقال المتغيرات بواسطة العناوين

وتكمن هذه الطريقة بإرسال عناوين المتغيرات إلى البرنامج المُنَادى . يعمل البرنامج المُنَادى إذاً على معطيات البرنامج المُنَادى . يبلغ البرنامج المُنَادى قيم المتغيرات بواسطة العنوان غير المباشرة . أي تعديل ، في البرنامج المُنَادى ، في قيمة منقولة ، معناه تعديل منطقة من البرنامج المُنَادى . هذه الطريقة هي نفسها المستعملة للإرسال بواسطة CALL CALL (SP name, arguments list) في فورتران . المخطط التالي يوضح لنا عملياً كيف أن متحوّلات البرنامج المُنَادى تصبح مركزية في البرنامج المُنَادى .



تُدعى متغيرات وهمية الرموز A ، B ، TAB الواردة في تعليمة النداء لأنها تتمتع فعلياً بقيمة معينة في لحظة النداء أو عند العودة .

تُدعى متغيرات شكلية الرموز U ، V ، W من SP التي ليست سوى أسماء تمثّل ، في لحظة النداء ، الرموز A ، B ، TAB من البرنامج المُنادي .

في لغة المؤول بإمكان المبرمج تصوّر عدة حلول لنقل المتغيرات إلى البرنامج المركزي ، فلنذكر البعض منها .

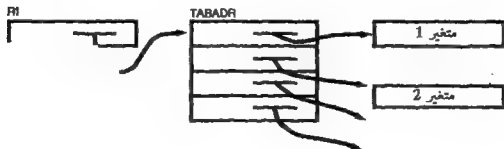
1- نضع المتغيرات في الجدول TAB ونرسل عنوان الجدول بواسطة أحد المرافف .

نداء	برنامج ثانوي
L R1,=A(TAB)	يتم بلوغ المتغير □ بواسطة
L R15,=A(SP)	L R4,DEPLAC(R1)
BALR R14,R15	أو بالتأشير
	L R4,0(R5,R1)
	وعندئذ يوضع المتغير بتصرفه في R4

2- نضع عنوان الجدول TAB مباشرة بعد تعليمة النداء

النداء	برنامج ثانوي
L R15,=A(SP)	R14 يسمح ببلوغ TAB .
CNOF 2,4 (تراصف)	العودة تتم بواسطة :
BALR R14,R15	BC 15,4(R14)
DC A(TAB)	

3- تكون المتغيرات عادة غير متراسة في البرنامج ونُفضّل عادة اعتماد التقنية المستعملة بواسطة المصرّقات . نقوم بإرسال عنوان الجدول الذي يحتوي على عناوين المتغيرات بواسطة أحد المرافف .



تعليمات		برنامج ثانوي	
L	R1:=A(TABADR)	WORK EQU ...	مرصف عمل
L	R15:=A(SP)		
BALR	R14,R15	L WORK,0(R1)	
		L WORK,0(WORK)	
		WORK المتغير الأول في:	
		L WORK,4(R1)	
		L WORK,0(WORK)	
		WORK المتغير الثاني في:	

هذا الحل هو المعتمد في لغة فورتران ، ويسمح ، في لغة المؤلف ، باستعادة المتغيرات المرسله بواسطة أحد البرامج فورتران وبالعكس .

نشير هنا إلى الفرق بين المتغيرات المرسله ومتغيرات العوده ، وهي تنتمي إلى البرنامج المُنَادِي . كما نفضل إستعمال مرصاف حسب نفس الاتفاقات المستعملة في أنظمة التشغيل (فقرة 4.21) . تسمح التعليمه CALL بإرسال من هذا النوع .

4.21 . إتفاقات الإتصال بين النظام والبرنامج

يبدأ التنفيذ منذ اللحظة التي يتم فيها إعداد عداد البرنامج وتخزين عنوان التعليمه الأولى للتنفيذ فيه . يقوم نظام التشغيل بهذه المهمه ، مما يفترض علينا إعتبار كل برنامج مستعمل كبرنامج ثانوي للنظام . من هنا فإن برنامج المستعمل يجب أن يبدأ بتمهيد يتعلّق بشروط إستعمال المرصاف من قِبل النظام .

تسمّى المرصاف 0 ، 1 ، 13 ، 14 و 15 مرصاف ربط «linkage registers» في وثائق المصمّم . وتستعمل بواسطة النظام والمرصّفات بشكل نموذجي وهذا هو السبب الذي من أجله يعتمد المستعمل على نفس الاتفاقات في الاتصالات مع البرامج الثانوية الخاصه به . في النظام OS ، يجب على البرنامج الثانوي أن يجمع مرصاف المُنَادِي في منطقه تدعى SAVE AREA ، تنتمي إلى البرنامج المُنَادِي . تجلّد تركيبة هذه المنطقه على الشكل التالي :

الكلمه المحتوى

- 1 تستعمل بواسطة اللغة PL/1
- 2 عنوان SAVE AREA الداخلي السابقه (الخاصه بالمُنَادِي) .
- 3 عنوان SAVE AREA التاليه (الخاصه بالمُنَادِي) .
- 4 عنوان العوده إلى المُنَادِي (مرصف 14) .
- 5 عنوان نقطه الدخول إلى البرنامج (مرصف 15) .
- 6 مرصف 0 .
- 7 مرصف 1 .

18 مرصف 12 .

عندما ينقل النظام التحكم إلى البرنامج :

- يحتوي المرصف 15 على عنوان نقطة الدخول إلى البرنامج . بإمكان البرنامج المنادي أن يشحن المرصف القاعدي الخاص به بواسطة التعليمة 15,LR REGBASE ، باعتبار نقطة الدخول وكأنها عنوان قاعدي .

- المرصف 14 يحتوي على عنوان العودة .

- المرصف 13 يحتوي على العنوان في SAVEAREA للبرنامج المنادي . نجد هنا شرح استعمال القاعدة 13 في التعليمة 13,12,12,STM الموجودة في جميع التمهيدات للبرامج .

- المرصف 1 يحتوي على عنوان جدول الكلمات التي تحتوي على عناوين المتغيرات الوهمية المتفولة . هذا الإتفاق يُستعمل ، مثلاً ، عندما يطلب برنامج فورتران برنامجاً آخر بلغة المؤول .

- المرصف 0 ، يستعمل ، عند العودة ، لإرسال نتيجة إحدى الدوال (مثلاً الدالة FUNCTION في فورتران) .

وبالتتبع ، ومنذ اللحظة التي يأخذ فيها البرنامج المنادي التحكم ويعود إلى التنفيذ ، فإنه :

- يُعرّف المنطقة الخاصة به SAVE AREA ،

- يُخزن مرصيف البرنامج المنادي بواسطة :

STM 14, 12, 12 (13)

في المنطقة SAVE AREA للمنادي

- يعرف مرصيف قاعلة ويشحن فيه قيمة معينة بواسطة :

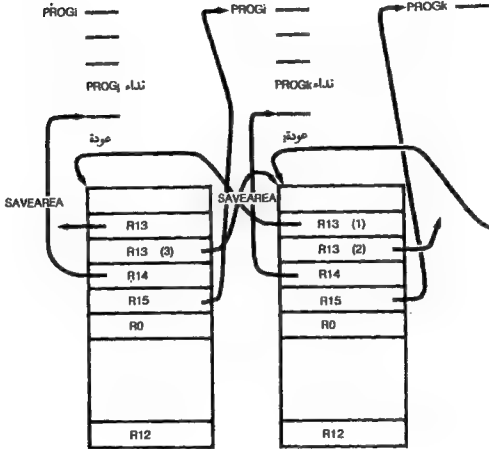
15,LR, 0 أو BALR

- يقوم بإجراء الوصلة بين المناطق SAVE AREA : ويُخزن ، في الكلمة الثانية من المنطقة SAVE AREA الخاصة به عنوان المنطقة الخاصة بالبرنامج المنادي (مرصيف 13) وفي الكلمة الثالثة من المنطقة SAVE AREA الخاصة بالمنادي ، عنوان المنطقة SAVE AREA الخاصة به .

- عند العودة ، فإن البرنامج المنادي يعيد تخزين مرصيف البرنامج المنادي مما يؤدي إلى العودة بواسطة 14,BR .

بإمكانه استعمال المرصف 15 لترميم كود العودة .

المخطط التالي يوضح عملية الربط بين المناطق SAVE AREA .



مخطط 1.21

ملاحظات : إذا كان البرنامج المنادي ، $PROGi$ ، مثلاً ، لا ينقل التحكم إلى برامج ثانوية أخرى كالبرنامج $PROGk$ ، فلا حاجة لتعريف $SAVE AREA$ لهذا البرنامج . من الواجب إذا السهر على حماية المرفص 13 الذي يسمح بإعادة مفهوم التنفيد الى البرنامج المنادي .

- (1) يتعلّق ذلك بالمرفص $R13$ من $PROGi$
- (2) يتعلّق ذلك بالمرفص $R13$ من $PROGk$
- (3) يتعلّق ذلك بالمرفص $R13$ من $PROGj$

LOC	OBJECT CODE	ADDR1	ADDR2	SYMT	SOURCE STATEMENT
700000				1	PROGJ DSCT ON
700010	90EC D00C	0000C		2	PROLOGUE SW 10,12,12(13)
700020	19CF	00020		3	DEFINITION ET CHARGEMENT DU REGISTRE DE BASE
700030	5000 C01C	0001C		4	USING PROGJ.12 LE REG 12 EST POUR BASE
700040	182D	00018		5	LR 10,12 DE R13 DANS LA SAVE AREA DE CE PROGRAMME.
700050	4100 C018	00018		6	SAUVEGARDE DE L'ADRESSE DE LA SA DE L'APPELANT
700060	5028 C008	00008		7	LR 10,12 DE R13 DANS LA SAVE AREA DE CE PROGRAMME.
700070	47F8 C080	00080		8	SAUVEGARDE DE L'ADRESSE DE LA SA DE L'APPELANT
700080	00018	00018		9	LR 2,13
700090	00018	00018		10	LR 1,24
700100	00018	00018		11	LR 1,24
700110	00018	00018		12	LR 1,24
700120	00018	00018		13	LR 1,24
700130	00018	00018		14	LR 1,24
700140	00018	00018		15	LR 1,24
700150	00018	00018		16	LR 1,24
700160	00018	00018		17	LR 1,24
700170	00018	00018		18	LR 1,24
700180	00018	00018		19	LR 1,24
700190	00018	00018		20	LR 1,24
700200	5870 C070	00070		21	SEQUENCE D'APPEL DE PROGK
700210	08EP			22	LR 10,12
700220				23	LR 10,12
700230				24	LR 10,12
700240				25	LR 10,12
700250				26	LR 10,12
700260				27	LR 10,12
700270				28	LR 10,12
700280				29	LR 10,12
700290				30	LR 10,12
700300				31	LR 10,12
700310				32	LR 10,12
700320				33	LR 10,12
700330				34	LR 10,12
700340				35	LR 10,12
700350				36	LR 10,12
700360				37	LR 10,12
700370				38	LR 10,12
700380				39	LR 10,12
700390				40	LR 10,12
700400				41	LR 10,12
700410				42	LR 10,12
700420				43	LR 10,12
700430				44	LR 10,12
700440				45	LR 10,12
700450				46	LR 10,12
700460				47	LR 10,12
700470				48	LR 10,12
700480				49	LR 10,12
700490				50	LR 10,12
700500				51	LR 10,12
700510				52	LR 10,12
700520				53	LR 10,12
700530				54	LR 10,12
700540				55	LR 10,12
700550				56	LR 10,12
700560				57	LR 10,12
700570				58	LR 10,12
700580				59	LR 10,12
700590				60	LR 10,12
700600				61	LR 10,12
700610				62	LR 10,12
700620				63	LR 10,12
700630				64	LR 10,12
700640				65	LR 10,12
700650				66	LR 10,12
700660				67	LR 10,12
700670				68	LR 10,12
700680				69	LR 10,12
700690				70	LR 10,12
700700				71	LR 10,12
700710				72	LR 10,12
700720				73	LR 10,12
700730				74	LR 10,12
700740				75	LR 10,12
700750				76	LR 10,12
700760				77	LR 10,12
700770				78	LR 10,12
700780				79	LR 10,12
700790				80	LR 10,12
700800				81	LR 10,12
700810				82	LR 10,12
700820				83	LR 10,12
700830				84	LR 10,12
700840				85	LR 10,12
700850				86	LR 10,12
700860				87	LR 10,12
700870				88	LR 10,12
700880				89	LR 10,12
700890				90	LR 10,12
700900				91	LR 10,12
700910				92	LR 10,12
700920				93	LR 10,12
700930				94	LR 10,12
700940				95	LR 10,12
700950				96	LR 10,12
700960				97	LR 10,12
700970				98	LR 10,12
700980				99	LR 10,12
700990				100	LR 10,12

EXTERNAL SYMBOL DICTIONARY

SYMBOL	TYPE	ID	ADDR	LENGTH	LDID
PROGJ	SD	0021	000300	000074	
PROGK	ER	0021	000300	000074	

التعليمة STM تسمح بترتيب مرادف متتالية عند كل رغبة باستعمال مرادف متجاورة .

إنفاقات الربط المعرفة سابقاً تسمح بطلبات المتابعة الداخلة ضمن البرامج . وهي لا تسمح أبداً بإجراء طلبات متتابعة تكرارية تحتاج إلى تعريف مكدس (STACK) خزن للنص . هذه الأولويات ليست موضوع هذا الكتاب . ولكن نشير إلى أن النظام OS يضع بتصرف المستعمل الوسائط لتعريف وإدارة منطقة من الذاكرة لكتابة برامج تكرارية (ماكرو GETMAIN) .

وللحاجة إلى التناسق والتوافق ، فإن المبرمج سيقوم بنفس عمليات الاختيار كالنظام OS في استعمال المرادف لإجراء الاتصالات بين البرامج الثانوية .

22 . التأويل المشروط وماكرو التعليمات

1.22 . التأويل المشروط

التأويل المشروط هو عبارة عن خطوة جديدة في التطور من لغة المكنة إلى اللغة المتطورة . ويتعلق ذلك بلغة تسمح بإنشاء وتوليد ، في مرحلة ما قبل التأويل ، نص مستهدف (object text) يمكن معالجته بواسطة المؤول . النص المؤول الناتج يمكن ، حسب القيم الأولية المخصصة لتحولات التأويل المشروط ، أن يتغير من تأويل إلى آخر . بإمكاننا مثلاً ، إدخال ، خلال مرحلة إعداد البرنامج ، متتالية من التعليمات (طباعة وسيطية تسمح بمتابعة أثر (trace) البرنامج) التي ، بواسطة تعديل بسيط للقيم الأولية لتحولات التأويل المشروط ، سيتم إلغاؤها عند التأويل النهائي . هذه العملية ، مضافة إلى استعمال الماكرو تعليمات⁽¹⁾ تجعل المؤول قريباً من اللغة المتطورة ، وتسمح للمبرمج بأن يُجهز بوسائل كالتعليمات : DO ... WHILE ، ... DO ، ... PERFORM التي تُسهّل البرمجة .

من غير الممكن هنا عرض جميع إمكانيات التأويل المشروط . سنحاول عرض الخطوط العريضة لهذه الطريقة بواسطة أمثلة توضح لنا العملية .

1.1.22 . متحولات وثوابت التأويل المشروط

التأويل المشروط يُعالج رموزاً بقيم قابلة للتغير : وهي عبارة عن متحولات التأويل . تبدأ أسماؤها بالرمز « & » ، وتحتوي على أكثر من ثمان سيات أبجعددية ، بما فيها « & » . السمة الثانية يجب أن تكون حرفاً . متحولات التأويل هي من ثلاثة أنواع A ، B و C أي حسابية ، منطقية وأبجعددية . يمكنها أن تكون مركزة بداخل ماكرو - إجراء والكود - المفتوح⁽²⁾ (Open-code) أو شاملة (كُلية) في جميع ماكرو - الاجراءات وفي الكود المفتوح . يجب أن يصرّح عن جميع متحولات التأويل ، المركزية

(1) مُصطلح معرف في 2.6

(2) الكود المفتوح : (Open code) قسم من كود المصدر يكون -موجوداً خارج وبعد الماكرو - تعريفات

والكَلِيَّة ، قبل إستعمالها . ويتمّ التصريح حسب نوع التحوّلة A ، B أو C :

LCLA ... LCLB ... LCLC ... (مركزية)
GBLA ... GBLB ... GBLC ... (كَلِيَّة)

لا يجب أن يُصرّح عن متغيرات الماكرو-تعريف (فقرة 1.2.22) . عند التصريح
توضع التحويلات A و B في صفر ويجري إعداد التحوّلة من النوع C في « سلسلة فارغة
من السّيات » .

لا يمكن بلوغ متحوّلة مُصرّح عنها عل أنها مركزية إلا في نفس الماكرو تعريف وفي
الكود المفتوح . أما التحوّلة المصّرّح عنها « شاملة (كَلِيَّة) » فيمكن بلوغها من ماكرو
تعريفات أخرى .

يمكن أن تكون متحوّلة التّأويل المشروط عبارة من متحوّلة مؤشّرة ، وفي هذه
الحالة يجب أن يتمّ التصريح عنها في مستوى LCLx أو GBLx كما نُصرّح عن الجدول
في فورتران . هكذا فإن :

LCLA &TAB(20)

تصرّح عن &TAB كجدول من 20 عنصراً نستطيع بلوغه بواسطة أحد
الأشكال التالية :

&TAB (تعبير حسابي) مثلاً : &TAB(&I+3)
&TAB (متحوّلة مؤشّرة) مثلاً : &TAB(&VAR(&I))

التعبير الذي يعطي قيمة المؤشر يجب أن يكون إيجابياً وأن لا يزيد عن حجم
الجدول المشار إليه في التصريح .

الثوابت الحسابية عبارة عن أعداد صحيحة بإشارة أو بدون إشارة حيث يجب أن
تكون قيمتها بين : $(2^{31} - 1 - 1 - 2^{31})$.

تأخذ الثوابت المنطقية القيمة 0 أو 1 التي تناسب الغلط والصحيح . الثوابت من
النوع سلسلة سيات تحتوي على عدد من 0 إلى 255 سمة محصورة بداخل فواصل عليا ،
ويمكن أن تكون مؤشّرة .
أمثلة :

(4) 'ABCDEF' ، تعادل 'D' ،

(2,3) 'ABCDEF' تعادل 'BCD' ،

المؤشر الأول يعطي الموقع الأولي للسلسلة الثانية والثاني يعطي طولها .

2.1.22 . أسماء الأوسمة

منطقة الرمز من أمر تأويل مشروط يمكن أن تحتوي على وسم تأويل مشروط . إنه عبارة عن رمز يبدأ بالنقطة « . » ويسمح ببلوغ أمر تأويل مشروط . لأسماء الوسم مدى مركزي .

3.1.22 . أوامر التخصيص SETA:

تقوم بتخصيص قيمة معينة إلى متحولة التأويل المشروط ، تتعلق بنوع المتحولات A ، B ، C وتتم بواسطة SETA ، SETB أو SETC . نشير إلى أن متحولة التأويل التي تحصل على التخصيص موجودة في المنطقة المحجوزة عادة للوسم . ولو افترضنا أن A ، B و C هي متحولات من النوع A ، B و C . نكتب :

منطقة الرمز	منطقة العملية	منطقة المعامل
AA	SETA	تعبير حسابي
AB	SETB	(تعبير منطقي)
AC	SETC	'تعبير أيجمدي'

وبشكل عام ، يحسب التعبير ويخزن القيمة الناتجة في متحولة التأويل الموجودة لجهة اليسار .

التعابير الحسابية

وتكتب بواسطة المؤثرات + ، - ، * و / (قسمة صحيحة بدون باق) . التقييم يتم من اليسار إلى اليمين بقواعد الأولوية العادية . أمثلة :

القيمة التي تأخذها المتحولة

AA1	SETA	10	10
AA1	SETA	AA1+1	11

التعابير المنطقية

تكتب بداخل أغلة بواسطة المؤثرات NOT ، AND و OR المذكورة في الترتيب التناقصي للأولويات . ويفضل وجود مؤثرات العلاقة بإمكاننا إجراء المقارنات بين التعابير الحسابية .

GT	GE	NE	BQ	LE	LT	مؤثرات علاقة :
>	≥	≠	=	≤	<	المعنى

يجب أن تكون المؤثرات محاطة بفراغات .

أمثلة :

BB4 SETB (BB1 OR BB2 AND BB3)
BB5 SETB (BA1 GT BA2)
BB6 SETB ('BC' EQ 'ALLOD')

تعايير من نوع سلسلة سيات

هي عبارة عن مجموعات من الثوابت والمتحولات من النوع الابعدي المحصورة بداخل قواصل عليا . المؤثر « . » (نقطة) يسمح بإجراء عمليات الإتحاد⁽¹⁾ . الترميز المؤشر يسمح باستخراج السلاسل الثانوية .

أمثلة :

		القيمة التي تأخذها التسمية
BC1	SETC 'CHA'	CHA
BC2	SETC 'BC1'	CHA
BC3	SETC 'BC1'.'ZNE'	CHAZNE
	ou 'BC1.ZNE'	
BC4	SETC 'CHAZNE'(2,5)	HAZNE
	الطول : الرتبة 1	
BC5	SETC 'BC4'(1,3)..'BC4'(5,1)	HAIE
BC6	SETC 'L'.'NON'	L'NON
BC7	SETC '5'	5 (caractère)
BC8	SETC 'BC7..25'	5.25 (un seul point)
BC9	SETC 'BA+10'	si BA = 10 alors
	ou 'BA.+10'	10+10 et non 20
BC10	SETC 'BC1BC1'	CHACHA
	ou 'BC1.BC1'	

نشير (BC10) إلى أن النقطة في عملية الإتحاد هي إختيارية عندما نجتمع بين متحولتين من السيات لأن الفاصل & لا يسمح بقيام أي نوع من الإجهام .

عندما تدخل المتحولات من النوع A إلى يمين الأمر (BC9) SETC ، فإن قيمة المتحولات تستبدل بالمتحولات ولكن بدون إجراء لأية عملية .

التعايير من النوع سلاسل السيات هي مهمة لأنها تسمح بإنشاء رموز أو بناء تعليقات لإتحاد متتالية . هناك أمثلة توضح إستعمالها عند دراسة الماكرو - إجراءات .

(1) عملية الربط - جمع سلسلتين ABCD و EF معناه تشكيل السلسلة ABCDEF .

بجعل التحولة &TEST تعادل صفرأ تكون قد ألغينا تأويل هذه التعليقات .

مثل 2

إنشاء نصّ معين .

التأويل للمشروط يمكن أن يستعمل لإنشاء نصّ متحوّل من تأويل إلى آخر . يمكن لهذا النصّ أن يكون رمزاً أو تعليمة .

يؤدي إلى توليد الأمر :

RAND	EQU	AND	
R1	EQU	1	تعادل 1

 إذا كانت التحولة AND تعادل 1

2.22 . الماكرو - إجراءات

باستعمال الماكرو إجراءات نجد أولية التأويل المشروط فالتعاطا .

الماكرو إجراء هو عبارة عن برنامج يعمل إسمأ مؤلفاً من سلسلة من التعليقات وأوامر التأويل المشروطة وغير المخصوصة بالأوامر MACRO و MEND .
 مثلاً : الماكرو تعريف التالي :

لائحة المتغيرات الشكلية إسم

	MACRO	
سطر مؤنثجي	SOMME	BU, BV, BW
جسم	{ L	1, BU
الإجرا	{ A	1, BV
	{ ST	1, BW
	MEND	

سيكون الماكرو تعريف موجوداً خارج البرنامج (open code) الذي يُراجعه .
 بإمكان الماكرو تعريف أن يكون موجوداً في مكتبة المستعمل أو مكتبة المؤول .
 الماكرو تعليقات هي إذاً السطر من البرنامج الذي يطلب من المؤول إدخال نصّ النموذج في البرنامج باستبدال المتغيرات الشكلية بالمتغيرات الفعلية .
 مثلاً :

يُولد التالية :

SOMME	A, B, C
L	1, A
A	1, B
ST	1, C

 المتغيرات الوسيطة الفعلية

نفترض عندئذٍ بأن هذا النظام ، المزود بالتأويل المشروط ، يسمح بإنشاء نماذج مستاندارد لبرامج يقوم المؤول بجعلها متوافقة مع كل حالة خاصة حسب قيم محاولات التأويل المشروط .

1.2.22 . تنقل المتغيرات

كما في حالة البرامج الثانوية ، المتغيرات الشكلية هي متغيرات السطر النموذجي في الماكرو تعريف والمتغيرات الفعلية هي متغيرات الماكرو تعلية . المتغيرات الشكلية هي رموز تسبقها السمة « & » .

يتكوّن السطر النموذجي في الماكرو تعريف على الشكل التالي :

اسم الإجراء	لائحة المتغيرات الشكلية
PROC	BU, &NO=3, &QTE=, &V, &RES=5, &H, &X

قيم نحو النقصان (0 أو سلسلة فارغة إن لم يجر تحديدها) .

المتغيرات الشكلية هي على نوعين :

- متغيرات الوضع : &X و &W, &V, &U في المثال ،
- متغيرات الكلمة المفتاح : &NO ، &QTE و &RES . وتمييزها بكون أسائها متبوعة بالرمز « = » وربّما بالقيمة التي تأخذها نحو النقصان ، قيمة تساوي « السلسلة الفارغة » في حال عدم تحديدها . ويتكوّن سطر نداء الماكرو تعريف كما يلي :

اسم الإجراء	لائحة المتغيرات الفعلية
PROC.	RES = 6, A, B, QTE = 4, D

- 1 - متغيرات مرتبطة بالمتغيرات الشكلية - من حيث مواقعها في اللائحة . لدينا هكذا التناسب بين A و &U ، B و &V ، D و &X . إن فاصلتين متاليتين تشيران إلى غياب متغير الوضع .
- 2 - متغيرات الكلمة المفتاح : الوصل بين المتغيرات الشكلية والفعلية القائم بفضل تشابه الاسم . هذه العناصر يجب أن يليها الرمز « = » وربّما قيمة تعدّل القيمة المحددة نحو النقصان . في مثلنا تأخذ RES القيمة 6 ، QTE القيمة 4 وتحفظ NO بالقيمة 3 نحو النقصان .

3 - قد تكون لوائح متغيرات معاطة بأهلة . لتأخذ الماكرو تعلية :

PROC 1 (A, B, C, D), K = (E, F, G, H)

والسطر النموذجي المناسب :
PROC 1 &POS, &K=

تتكون التغيرات الفعلية بواسطة اللاحقين (A, B, C, D) و (E, F, G, H).
 أما POS(3) فيستبدل عندئذٍ بـ C خلال انتشار الماكرو تعليمية . كذلك يُستبدل
 (2) K بـ F . بإمكان لوائح التغيرات أن تكون ذات أطوال متغيرة ، وسنرى أن
 الخاصية POS & N' تسمح بمعرفة طول اللائحة المرتبطة بـ POS .

2.2.22 . تطبيق

المثال التالي يقوم بتوليد تعليمات تسمح بجمع N خلية من الذاكرة منقولة إلى ماكرو
 الإجراء بواسطة لائحة RES ستحتوي على النتيجة و NB تمثل عدد العناصر
 المطلوب جمعها . المؤشر المركزي I يستعمل لمراجعة مختلف عناصر اللائحة .

```

1      MACRO
2      SOMME &MEN,&RES,&NB=&, &REG=
3      LCLA  &I
4      L     &REG,&MEN(1)
5      SETA 1
6      &BOUCLE
7      &I
8      SETA &I+1
9      AIF   (&I GT &NB),FIN
10     A     &REG,&MEN(&I)
11     AGO   &BOUCLE
12     ST    &REG,&RES
      PEND

```

```

000060 5030 C07A 00074 64      SOMME (A,B,C,D),X,NB=4,REG=3
000064 5A30 C07B 00078 65+     I 3=A
000068 5A30 C07C 0007C 66+     A 3=B
00006C 5A30 C080 00080 67+     A 3=C
000070 5030 C084 00084 68+     A 3=D
                                69+     ST 3,X

```

```

000074 72 A      DS F
000078 73 B      DS F
00007C 74 C      DS F
000080 75 D      DS F
000084 76 X      DS F

```

3.2.22 . الأمر MEXIT

يسمح بوقف تأويل الماكرو تعريف . من الممكن اعتباره معادلاً للتفريع إلى الأمر
 MEND .

4.2.22 . الأمر ACTR

يسمح بمراقبة عدد AIF و AGO الجاري خلال التأويل المشروط . ويكتب :
 ACTR (تعبير حسابي)

يؤدي إلى توليد عداد يعادل مضمونه قيمة التعبير الحسابي . يمكن أن يكون العداد
 مركزياً للماكرو تعريف أو شاملاً . في كل مرة يجري فيها تنفيذ AIF أو AGO بواسطة
 المؤول ، فإن العداد المناسب لهذا القسم من البرنامج يحقّض واحداً من
 مضمونه . وعندما يبلغ الصفر ، فإن المؤول يخرج من الماكرو تعريف (فعل معادل لـ

VIEXIT) أو يُوقف التأويل إذا كان ذلك متعلقاً بعدد شامل . هذا الأمر يسمح بتحديد عدد الحلقات التي تجري في مرحلة ما قبل التأويل .

5.2.22 . الأمر MNOTE

يمكن أن يستعمل من قبل المبرمج لتوليد رسالة الخطأ الخاصة به أو طباعة قيم وسيطة مأخوذة من متحولات التأويل .

ويمكن أن يكتب بعلّة أشكال :
وسم تأويل

- | | | | |
|-----|---------------------------|-------|-----------------|
| (1) | etiquette
d'assemblage | MNOTE | code, 'message' |
| (2) | étiq. assem. | MNOTE | 'message' |
| (3) | étiq. assem. | MNOTE | *, 'message' |
| (4) | étiq. assem. | MNOTE | 'message' |

الكود هو عبارة عن تعبير حسابي بقيمة محصورة بين 0 و 255 يربط مستوى من الخطأ بالرسالة . في الشكل 2 يفترض بالكود أن يكون مُعادلاً لـ 1 . لا تُطبع الرسالة من ضمن رسائل الخطأ إلا إذا كان الكود الذي يشير إلى درجة الحقيقة هو أعلى من أو يعادل الكود المعتمد من المؤلف .

الشكلان 3 و 4 يوردان الرسالة كمجرد ملاحظة . .

6.2.22 . الملاحظات :

من الممكن إدخال ملاحظات في ماكرو التعريفات على الشكل التالي :

* COMMENTAIRE GENRE

.* COMMENTAIRE NON GENRE

7.2.22 . الدوال من النوع الذاتي (Intrinsic)

&SYSLIST

تسمح ، داخل الماكرو تعريف ، بتسمية متغيرات الموقع الموجودة داخل ماكرو تعليمة النداء . وتكتب بمؤشر أو بمؤشرين يمكن أن يكونا عبارة عن تعابير حسابية من نوع ذلك الذي رأيناه في الفقرة 3.1.22 . سنختبر إستعمالها بالخاصية N° .

&SYSLIST(I) تشير إلى المتغير الفعلي الخاص بالموقع رقم i من التعليمة . يمكن أن يكون هذا المتغير الفعلي عبارة عن لائحة (حسب الفقرة 3.1.22). في هذه الحالة ، سنسمي العنصر رقم j من اللائحة بالرتبة I بواسطة &SYSLIST(I,J) . في المثال المذكور في الفقرة 2.2.22 &SYSLIST(1,2) تعني المتغير B ، و &SYSLIST(2) تعني X .

&SYSLIST(0) تعني الوسم الموجود قبل الماكرو تعليمة الخاصة بالنداء . هذه المهمة تسمح بتفادي تسمية المتغيرات .

&SYSNDX

هي عبارة عن عداد من أربعة أرقام عشرية ، وهو مركزي ضمن ماكرو - تعريف ، وتزداد قيمته عند كل استعمال جديد للماكرو . لا يمكن أن يُستعمل وحيداً ولكن يمكن أن يتحد مع رمزا . هذه هي الوسيلة لتوليد أوسمة مختلفة عند كل نداء للماكرو - التعريف وتسمح بتفادي الأخطاء في التأويل والناجئة عن تعريف الرموز .
مثلاً :

لفترض الماكرو - تعريف التالي :

```
MACRO
PROC  &A,...
AA&SYSNDX
.....
R&SYSNDX
.....
MEND
```

النداء الأول يتم بواسطة PROC ETIQ,...

المتحولة &A&SYSNDX تأخذ القيمة ETIQ0001

المتحولة R&SYSNDX تأخذ القيمة R0001

في النداء الثاني بواسطة ... PROC ETIQ,...

المتحولة &A&SYSNDX تأخذ القيمة ETIQ0002

المتحولة R&SYSNDX تأخذ القيمة R0002

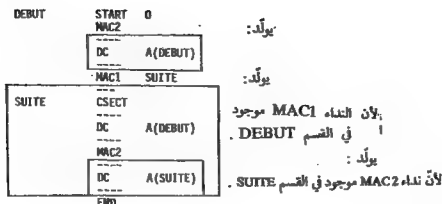
&SYSECT

تسمح بتعريف اسم القسم حيث توجد الماكرو - تعليمة للتأدية . المثل التالي يوضح

ذلك :

```
MACRO
MAC1  &ETIQ
.....
CSECT
.....
DC    A(&SYSECT)
.....
MAC2
.....

MEND
MACRO
MAC2
.....
DC    A(&SYSECT)
.....
MEND
```



&SYSPARM

يعطي وسيلة الرجوع إلى المتغير SYSPARM لبطاقة //EXEC (Job JCL في Control Language : لغة مراقبة العمل) .
 مثلاً .

```
// EXEC ASMC,PARM=SYSPARM(DEBUG)
//ASM.SYSIN DD *
TEST START 0
-----
AIF ('&SYSPARM' NE 'DEBUG'). (قفزة)
-----
ولادة تعليقات
تنفيذ وتقويم
.SAINT ANOP
```

&SYSTIME

يعطي ساعة التأويل بواسطة خمس سيات : h.h.mm

&SYSDATE

يعطي التاريخ بواسطة ثمان سيات : mm/jj/aa

8.2.22 . الخصائص

مفهوم الخاصية المرتبط بمعطى أو بتعليمة جرت إثارته في الفقرة 2.3.6 .. كما

إستعملنا الخاصية - طول (فقرة 3.2.7) . يسمح للزول لنا باستعمال خاصيات أخرى حيث البعض منها يجد إستعمالاً بسبب وجود إمكانيات التأويل المشروط .

الخاصية : TYPE T°

وقيمتها سمة أبجدية حسب نوع الرمز المطبقة عليه . إذا كانت NUM ، مثلاً ، عبارة عن ثابتة عشرية . موسعة ؛ فإن قيمة T°NUM ستكون Z . الحرف الذي يُميز النوع هو نفسه المستعمل في الأوامر A:DC تناسب ثابتة عنوان من نوع A ، بينما B تناسب ثابتة منطقية ... ونضيف التناسبات التالية :

G	ثابتة بفاصلة ثابتة وطول محدد ظاهر
K	ثابتة بفاصلة متحركة وطول عدد ظاهر
R	ثابتة عنوان بطول محدد ظاهر
I	تعليمة - آلة
M	ماكرو تعليمة
W	CCW
J	اسم قسم
T	رمز خارجي
N	قيمة تعريف أوتوماتيكي
O	سمة مخلوطة

{ تتعلّقان بمتغيرات الماكرو تعليمة

الخاصية L° LONGUEUR (طول)

جرت دراستها في الفقرة 3.2.7 .

الخاصية مقياس S°

عبارة عن قيمة رقمية تتعلّق بنوع الرمز .

- لعدد عشري (نوع P أو Z)

عبارة عن عدد الأرقام في القسم الكسري .

- لعدد بفاصلة متحركة (أنواع L, E, D أو K)

إنه عدد الأصفار السادس عشرية في يسار القسم العشري (الوزن الأكبر) .

- لعدد بفاصلة ثابتة (الأنواع M ، F أو G)

عبارة عن القوة 2 التي يتم ضرب قيمة الثابتة بها . وتشير إلى عدد البتات في

القسم الكسري إذا كان إيجابياً ، وعدد البتات المتروكة إذا كان سلبياً .

الخاصية قسم صحيح I'

عبارة عن عدد يتعلّق بـ S' و L' .

$I' = 2 * L' - S' - 1$ لعدد عشري من نوع P

$I' = L' - S'$ لعدد عشري من نوع Z

$L' \leq 8$ مع $I' = 2 * (L' - 1) - S$ لعدد بفاصلة متحركة من نوع

K , L , E , D

$L' > 8$ مع $I' = 2 * (L' - 1) - S' - 2$ لعدد بفاصلة متحركة من نوع L

$I' = 8 * L' - S' - 1$ لعدد بفاصلة ثابتة من نوع

G , F , H

الخاصية عدد السيات K'

وتُطبّق فقط على مُتغيّرات الماكرو - تعليمة وأيضاً ، بإشراف OS ، على الرموز المتحوّلة .. & وعلّ ألدوال الذاتية (من نوع intrinsic) . وتعطي عدد سيات الرمز التي تطبّق عليه .

أمثلة : في مثل الفقرة 2.2.22 : $K' \& MEM = 9$

$\&A \text{ SETA } 253 : K' \& A = 3,$

$\&B \text{ SETB } 0 : K' \& B = 1,$

$\&C \text{ SETC 'ALPHA' : } K' \& C = 5.$

الخاصية عدد العناصر من اللائحة N'

وتتطبّق فقط على مُتغيّرات الماكرو - تعليمة ، وتعطي عدد عناصر اللائحة .
مثلاً :

$\text{PROC } \&A, \&B, \&K =$
 $\text{PROC } (1, 2, 4), U, K = 3$

خط نموذج

ماكرو تعليمة

$N' \& A = 4$
 $N' \& SYSLIST = 2$
 $N' \& SYSLIST(1) = 4.$

(يتمّ تعداد السيات غير الموجودة)
مُتغيّرات الموقع

9.2.22 . أمثلة عن الماكرو - تعريفات

الماكرو - تعريف التالي يسمح بتوليد الأوامر (التوجيهات) المُعادلة لـ $R \& EQUI$

12	PACRO	
13	EQUREG	
14	MACRO D'EQUIVALENCE REGISTRES	ماكرو معادل للرافص
15	GBLA AND	
16	AND SETA I	
17	.ST AIF {AND GT IS}.FIN	
18	RAND EQU AND	
19	AND SETA AND+1	
20	AGO +S1	
21	.FIN MEND	

وُيُولد الكود التالي :

	33	EQU REG	
	34**	MACRO D	ماكرو معادل للمرافف
00001	35+R1	EQU 1	
00002	36+R2	EQU 2	
00003	37+R3	EQU 3	
00004	38+R4	EQU 4	
00005	39+R5	EQU 5	
00006	40+R6	EQU 6	
00007	41+R7	EQU 7	
00008	42+R8	EQU 8	
00009	43+R9	EQU 9	
0000A	44+R10	EQU 10	
0000B	45+R11	EQU 11	
0000C	46+R12	EQU 12	
0000D	47+R13	EQU 13	
0000E	48+R14	EQU 14	
0000F	49+R15	EQU 15	

الماكرو- تعريف PROLOGUE يسمح بشحن واحد أو عدة مرافف قاعدة مخزناً مفهوم البرنامج المتادي حسب المعايير العادية المُحددة في الفصل 21 . وهو يُعرّف في نفس الوقت منطقة SAVE AREA للبرنامج الجاري . عنوان القاعدة الذي جرى اختياره هو عنوان نقطة الدخول إلى البرنامج . ويولد الكود المُولد على الصفحة التالية .

```

14      MACRO
15      PROLOGUE &ABASE=&ABASE=
16      EQU 0
17      LCLA  &I,&O
18      LCLC  &CH
19      SYN  14,12,12(13)
20      *
21      &CH  SETC  '&ABASE(1)'
22      &I   SETA  2
23      &T1  AIF  (&I GT N*&ABASE).SUIT1
24      &CH  SETC  '&CH*.*.*&ABASE(&I)'
25      &I   SETA  &I+1
26      AGO  &T1
27      *SUIT1
28      ANOP
29      USING &ABASE,&CH
30      LR   &ABASE(1),15
31      LR   13,SAVEAREA+4
32      LR   2,13
33      LA   13,SAVEAREA
34      ST   13,B(2)
35      *
36      CONSTRUCTION DES CHARGEMENTS DES REGISTRES DE BASE
37      &I   SETA  2
38      &T2  AIF  (&I GT N*&ABASE).SUIT2
39      L    0,=F'4096'
40      &T3  AIF  (&I GT N*&ABASE).SUIT2
41      LR   15,0
42      LR   &ABASE(&I),15
43      &I   SETA  &I+1
44      AGO  &T3
45      *SUIT2
46      ANOP
47      B    0+76
48      DS   1BF
49      *
50      ENDBLOCK
51      ENDBLOCK
52      ENDBLOCK
53      ENDBLOCK
54      ENDBLOCK
55      ENDBLOCK
56      ENDBLOCK
57      ENDBLOCK
58      ENDBLOCK
59      ENDBLOCK
60      ENDBLOCK
61      ENDBLOCK
62      ENDBLOCK
63      ENDBLOCK
64      ENDBLOCK
65      ENDBLOCK
66      ENDBLOCK
67      ENDBLOCK
68      ENDBLOCK
69      ENDBLOCK
70      ENDBLOCK
71      ENDBLOCK
72      ENDBLOCK
73      ENDBLOCK
74      ENDBLOCK
75      ENDBLOCK
76      ENDBLOCK
77      ENDBLOCK
78      ENDBLOCK
79      ENDBLOCK
80      ENDBLOCK
81      ENDBLOCK
82      ENDBLOCK
83      ENDBLOCK
84      ENDBLOCK
85      ENDBLOCK
86      ENDBLOCK
87      ENDBLOCK
88      ENDBLOCK
89      ENDBLOCK
90      ENDBLOCK
91      ENDBLOCK
92      ENDBLOCK
93      ENDBLOCK
94      ENDBLOCK
95      ENDBLOCK
96      ENDBLOCK
97      ENDBLOCK
98      ENDBLOCK
99      ENDBLOCK
100     ENDBLOCK
101     ENDBLOCK
102     ENDBLOCK
103     ENDBLOCK
104     ENDBLOCK
105     ENDBLOCK
106     ENDBLOCK
107     ENDBLOCK
108     ENDBLOCK
109     ENDBLOCK
110     ENDBLOCK
111     ENDBLOCK
112     ENDBLOCK
113     ENDBLOCK
114     ENDBLOCK
115     ENDBLOCK
116     ENDBLOCK
117     ENDBLOCK
118     ENDBLOCK
119     ENDBLOCK
120     ENDBLOCK
121     ENDBLOCK
122     ENDBLOCK
123     ENDBLOCK
124     ENDBLOCK
125     ENDBLOCK
126     ENDBLOCK
127     ENDBLOCK
128     ENDBLOCK
129     ENDBLOCK
130     ENDBLOCK
131     ENDBLOCK
132     ENDBLOCK
133     ENDBLOCK
134     ENDBLOCK
135     ENDBLOCK
136     ENDBLOCK
137     ENDBLOCK
138     ENDBLOCK
139     ENDBLOCK
140     ENDBLOCK
141     ENDBLOCK
142     ENDBLOCK
143     ENDBLOCK
144     ENDBLOCK
145     ENDBLOCK
146     ENDBLOCK
147     ENDBLOCK
148     ENDBLOCK
149     ENDBLOCK
150     ENDBLOCK
151     ENDBLOCK
152     ENDBLOCK
153     ENDBLOCK
154     ENDBLOCK
155     ENDBLOCK
156     ENDBLOCK
157     ENDBLOCK
158     ENDBLOCK
159     ENDBLOCK
160     ENDBLOCK
161     ENDBLOCK
162     ENDBLOCK
163     ENDBLOCK
164     ENDBLOCK
165     ENDBLOCK
166     ENDBLOCK
167     ENDBLOCK
168     ENDBLOCK
169     ENDBLOCK
170     ENDBLOCK
171     ENDBLOCK
172     ENDBLOCK
173     ENDBLOCK
174     ENDBLOCK
175     ENDBLOCK
176     ENDBLOCK
177     ENDBLOCK
178     ENDBLOCK
179     ENDBLOCK
180     ENDBLOCK
181     ENDBLOCK
182     ENDBLOCK
183     ENDBLOCK
184     ENDBLOCK
185     ENDBLOCK
186     ENDBLOCK
187     ENDBLOCK
188     ENDBLOCK
189     ENDBLOCK
190     ENDBLOCK
191     ENDBLOCK
192     ENDBLOCK
193     ENDBLOCK
194     ENDBLOCK
195     ENDBLOCK
196     ENDBLOCK
197     ENDBLOCK
198     ENDBLOCK
199     ENDBLOCK
200     ENDBLOCK
201     ENDBLOCK
202     ENDBLOCK
203     ENDBLOCK
204     ENDBLOCK
205     ENDBLOCK
206     ENDBLOCK
207     ENDBLOCK
208     ENDBLOCK
209     ENDBLOCK
210     ENDBLOCK
211     ENDBLOCK
212     ENDBLOCK
213     ENDBLOCK
214     ENDBLOCK
215     ENDBLOCK
216     ENDBLOCK
217     ENDBLOCK
218     ENDBLOCK
219     ENDBLOCK
220     ENDBLOCK
221     ENDBLOCK
222     ENDBLOCK
223     ENDBLOCK
224     ENDBLOCK
225     ENDBLOCK
226     ENDBLOCK
227     ENDBLOCK
228     ENDBLOCK
229     ENDBLOCK
230     ENDBLOCK
231     ENDBLOCK
232     ENDBLOCK
233     ENDBLOCK
234     ENDBLOCK
235     ENDBLOCK
236     ENDBLOCK
237     ENDBLOCK
238     ENDBLOCK
239     ENDBLOCK
240     ENDBLOCK
241     ENDBLOCK
242     ENDBLOCK
243     ENDBLOCK
244     ENDBLOCK
245     ENDBLOCK
246     ENDBLOCK
247     ENDBLOCK
248     ENDBLOCK
249     ENDBLOCK
250     ENDBLOCK
251     ENDBLOCK
252     ENDBLOCK
253     ENDBLOCK
254     ENDBLOCK
255     ENDBLOCK
256     ENDBLOCK
257     ENDBLOCK
258     ENDBLOCK
259     ENDBLOCK
260     ENDBLOCK
261     ENDBLOCK
262     ENDBLOCK
263     ENDBLOCK
264     ENDBLOCK
265     ENDBLOCK
266     ENDBLOCK
267     ENDBLOCK
268     ENDBLOCK
269     ENDBLOCK
270     ENDBLOCK
271     ENDBLOCK
272     ENDBLOCK
273     ENDBLOCK
274     ENDBLOCK
275     ENDBLOCK
276     ENDBLOCK
277     ENDBLOCK
278     ENDBLOCK
279     ENDBLOCK
280     ENDBLOCK
281     ENDBLOCK
282     ENDBLOCK
283     ENDBLOCK
284     ENDBLOCK
285     ENDBLOCK
286     ENDBLOCK
287     ENDBLOCK
288     ENDBLOCK
289     ENDBLOCK
290     ENDBLOCK
291     ENDBLOCK
292     ENDBLOCK
293     ENDBLOCK
294     ENDBLOCK
295     ENDBLOCK
296     ENDBLOCK
297     ENDBLOCK
298     ENDBLOCK
299     ENDBLOCK
300     ENDBLOCK
301     ENDBLOCK
302     ENDBLOCK
303     ENDBLOCK
304     ENDBLOCK
305     ENDBLOCK
306     ENDBLOCK
307     ENDBLOCK
308     ENDBLOCK
309     ENDBLOCK
310     ENDBLOCK
311     ENDBLOCK
312     ENDBLOCK
313     ENDBLOCK
314     ENDBLOCK
315     ENDBLOCK
316     ENDBLOCK
317     ENDBLOCK
318     ENDBLOCK
319     ENDBLOCK
320     ENDBLOCK
321     ENDBLOCK
322     ENDBLOCK
323     ENDBLOCK
324     ENDBLOCK
325     ENDBLOCK
326     ENDBLOCK
327     ENDBLOCK
328     ENDBLOCK
329     ENDBLOCK
330     ENDBLOCK
331     ENDBLOCK
332     ENDBLOCK
333     ENDBLOCK
334     ENDBLOCK
335     ENDBLOCK
336     ENDBLOCK
337     ENDBLOCK
338     ENDBLOCK
339     ENDBLOCK
340     ENDBLOCK
341     ENDBLOCK
342     ENDBLOCK
343     ENDBLOCK
344     ENDBLOCK
345     ENDBLOCK
346     ENDBLOCK
347     ENDBLOCK
348     ENDBLOCK
349     ENDBLOCK
350     ENDBLOCK
351     ENDBLOCK
352     ENDBLOCK
353     ENDBLOCK
354     ENDBLOCK
355     ENDBLOCK
356     ENDBLOCK
357     ENDBLOCK
358     ENDBLOCK
359     ENDBLOCK
360     ENDBLOCK
361     ENDBLOCK
362     ENDBLOCK
363     ENDBLOCK
364     ENDBLOCK
365     ENDBLOCK
366     ENDBLOCK
367     ENDBLOCK
368     ENDBLOCK
369     ENDBLOCK
370     ENDBLOCK
371     ENDBLOCK
372     ENDBLOCK
373     ENDBLOCK
374     ENDBLOCK
375     ENDBLOCK
376     ENDBLOCK
377     ENDBLOCK
378     ENDBLOCK
379     ENDBLOCK
380     ENDBLOCK
381     ENDBLOCK
382     ENDBLOCK
383     ENDBLOCK
384     ENDBLOCK
385     ENDBLOCK
386     ENDBLOCK
387     ENDBLOCK
388     ENDBLOCK
389     ENDBLOCK
390     ENDBLOCK
391     ENDBLOCK
392     ENDBLOCK
393     ENDBLOCK
394     ENDBLOCK
395     ENDBLOCK
396     ENDBLOCK
397     ENDBLOCK
398     ENDBLOCK
399     ENDBLOCK
400     ENDBLOCK
401     ENDBLOCK
402     ENDBLOCK
403     ENDBLOCK
404     ENDBLOCK
405     ENDBLOCK
406     ENDBLOCK
407     ENDBLOCK
408     ENDBLOCK
409     ENDBLOCK
410     ENDBLOCK
411     ENDBLOCK
412     ENDBLOCK
413     ENDBLOCK
414     ENDBLOCK
415     ENDBLOCK
416     ENDBLOCK
417     ENDBLOCK
418     ENDBLOCK
419     ENDBLOCK
420     ENDBLOCK
421     ENDBLOCK
422     ENDBLOCK
423     ENDBLOCK
424     ENDBLOCK
425     ENDBLOCK
426     ENDBLOCK
427     ENDBLOCK
428     ENDBLOCK
429     ENDBLOCK
430     ENDBLOCK
431     ENDBLOCK
432     ENDBLOCK
433     ENDBLOCK
434     ENDBLOCK
435     ENDBLOCK
436     ENDBLOCK
437     ENDBLOCK
438     ENDBLOCK
439     ENDBLOCK
440     ENDBLOCK
441     ENDBLOCK
442     ENDBLOCK
443     ENDBLOCK
444     ENDBLOCK
445     ENDBLOCK
446     ENDBLOCK
447     ENDBLOCK
448     ENDBLOCK
449     ENDBLOCK
450     ENDBLOCK
451     ENDBLOCK
452     ENDBLOCK
453     ENDBLOCK
454     ENDBLOCK
455     ENDBLOCK
456     ENDBLOCK
457     ENDBLOCK
458     ENDBLOCK
459     ENDBLOCK
460     ENDBLOCK
461     ENDBLOCK
462     ENDBLOCK
463     ENDBLOCK
464     ENDBLOCK
465     ENDBLOCK
466     ENDBLOCK
467     ENDBLOCK
468     ENDBLOCK
469     ENDBLOCK
470     ENDBLOCK
471     ENDBLOCK
472     ENDBLOCK
473     ENDBLOCK
474     ENDBLOCK
475     ENDBLOCK
476     ENDBLOCK
477     ENDBLOCK
478     ENDBLOCK
479     ENDBLOCK
480     ENDBLOCK
481     ENDBLOCK
482     ENDBLOCK
483     ENDBLOCK
484     ENDBLOCK
485     ENDBLOCK
486     ENDBLOCK
487     ENDBLOCK
488     ENDBLOCK
489     ENDBLOCK
490     ENDBLOCK
491     ENDBLOCK
492     ENDBLOCK
493     ENDBLOCK
494     ENDBLOCK
495     ENDBLOCK
496     ENDBLOCK
497     ENDBLOCK
498     ENDBLOCK
499     ENDBLOCK
500     ENDBLOCK
501     ENDBLOCK
502     ENDBLOCK
503     ENDBLOCK
504     ENDBLOCK
505     ENDBLOCK
506     ENDBLOCK
507     ENDBLOCK
508     ENDBLOCK
509     ENDBLOCK
510     ENDBLOCK
511     ENDBLOCK
512     ENDBLOCK
513     ENDBLOCK
514     ENDBLOCK
515     ENDBLOCK
516     ENDBLOCK
517     ENDBLOCK
518     ENDBLOCK
519     ENDBLOCK
520     ENDBLOCK
521     ENDBLOCK
522     ENDBLOCK
523     ENDBLOCK
524     ENDBLOCK
525     ENDBLOCK
526     ENDBLOCK
527     ENDBLOCK
528     ENDBLOCK
529     ENDBLOCK
530     ENDBLOCK
531     ENDBLOCK
532     ENDBLOCK
533     ENDBLOCK
534     ENDBLOCK
535     ENDBLOCK
536     ENDBLOCK
537     ENDBLOCK
538     ENDBLOCK
539     ENDBLOCK
540     ENDBLOCK
541     ENDBLOCK
542     ENDBLOCK
543     ENDBLOCK
544     ENDBLOCK
545     ENDBLOCK
546     ENDBLOCK
547     ENDBLOCK
548     ENDBLOCK
549     ENDBLOCK
550     ENDBLOCK
551     ENDBLOCK
552     ENDBLOCK
553     ENDBLOCK
554     ENDBLOCK
555     ENDBLOCK
556     ENDBLOCK
557     ENDBLOCK
558     ENDBLOCK
559     ENDBLOCK
560     ENDBLOCK
561     ENDBLOCK
562     ENDBLOCK
563     ENDBLOCK
564     ENDBLOCK
565     ENDBLOCK
566     ENDBLOCK
567     ENDBLOCK
568     ENDBLOCK
569     ENDBLOCK
570     ENDBLOCK
571     ENDBLOCK
572     ENDBLOCK
573     ENDBLOCK
574     ENDBLOCK
575     ENDBLOCK
576     ENDBLOCK
577     ENDBLOCK
578     ENDBLOCK
579     ENDBLOCK
580     ENDBLOCK
581     ENDBLOCK
582     ENDBLOCK
583     ENDBLOCK
584     ENDBLOCK
585     ENDBLOCK
586     ENDBLOCK
587     ENDBLOCK
588     ENDBLOCK
589     ENDBLOCK
590     ENDBLOCK
591     ENDBLOCK
592     ENDBLOCK
593     ENDBLOCK
594     ENDBLOCK
595     ENDBLOCK
596     ENDBLOCK
597     ENDBLOCK
598     ENDBLOCK
599     ENDBLOCK
600     ENDBLOCK
601     ENDBLOCK
602     ENDBLOCK
603     ENDBLOCK
604     ENDBLOCK
605     ENDBLOCK
606     ENDBLOCK
607     ENDBLOCK
608     ENDBLOCK
609     ENDBLOCK
610     ENDBLOCK
611     ENDBLOCK
612     ENDBLOCK
613     ENDBLOCK
614     ENDBLOCK
615     ENDBLOCK
616     ENDBLOCK
617     ENDBLOCK
618     ENDBLOCK
619     ENDBLOCK
620     ENDBLOCK
621     ENDBLOCK
622     ENDBLOCK
623     ENDBLOCK
624     ENDBLOCK
625     ENDBLOCK
626     ENDBLOCK
627     ENDBLOCK
628     ENDBLOCK
629     ENDBLOCK
630     ENDBLOCK
631     ENDBLOCK
632     ENDBLOCK
633     ENDBLOCK
634     ENDBLOCK
635     ENDBLOCK
636     ENDBLOCK
637     ENDBLOCK
638     ENDBLOCK
639     ENDBLOCK
640     ENDBLOCK
641     ENDBLOCK
642     ENDBLOCK
643     ENDBLOCK
644     ENDBLOCK
645     ENDBLOCK
646     ENDBLOCK
647     ENDBLOCK
648     ENDBLOCK
649     ENDBLOCK
650     ENDBLOCK
651     ENDBLOCK
652     ENDBLOCK
653     ENDBLOCK
654     ENDBLOCK
655     ENDBLOCK
656     ENDBLOCK
657     ENDBLOCK
658     ENDBLOCK
659     ENDBLOCK
660     ENDBLOCK
661     ENDBLOCK
662     ENDBLOCK
663     ENDBLOCK
664     ENDBLOCK
665     ENDBLOCK
666     ENDBLOCK
667     ENDBLOCK
668     ENDBLOCK
669     ENDBLOCK
670     ENDBLOCK
671     ENDBLOCK
672     ENDBLOCK
673     ENDBLOCK
674     ENDBLOCK
675     ENDBLOCK
676     ENDBLOCK
677     ENDBLOCK
678     ENDBLOCK
679     ENDBLOCK
680     ENDBLOCK
681     ENDBLOCK
682     ENDBLOCK
683     ENDBLOCK
684     ENDBLOCK
685     ENDBLOCK
686     ENDBLOCK
687     ENDBLOCK
688     ENDBLOCK
689     ENDBLOCK
690     ENDBLOCK
691     ENDBLOCK
692     ENDBLOCK
693     ENDBLOCK
694     ENDBLOCK
695     ENDBLOCK
696     ENDBLOCK
697     ENDBLOCK
698     ENDBLOCK
699     ENDBLOCK
700     ENDBLOCK
701     ENDBLOCK
702     ENDBLOCK
703     ENDBLOCK
704     ENDBLOCK
705     ENDBLOCK
706     ENDBLOCK
707     ENDBLOCK
708     ENDBLOCK
709     ENDBLOCK
710     ENDBLOCK
711     ENDBLOCK
712     ENDBLOCK
713     ENDBLOCK
714     ENDBLOCK
715     ENDBLOCK
716     ENDBLOCK
717     ENDBLOCK
718     ENDBLOCK
719     ENDBLOCK
720     ENDBLOCK
721     ENDBLOCK
722     ENDBLOCK
723     ENDBLOCK
724     ENDBLOCK
725     ENDBLOCK
726     ENDBLOCK
727     ENDBLOCK
728     ENDBLOCK
729     ENDBLOCK
730     ENDBLOCK
731     ENDBLOCK
732     ENDBLOCK
733     ENDBLOCK
734     ENDBLOCK
735     ENDBLOCK
736     ENDBLOCK
737     ENDBLOCK
738     ENDBLOCK
739     ENDBLOCK
740     ENDBLOCK
741     ENDBLOCK
742     ENDBLOCK
743     ENDBLOCK
744     ENDBLOCK
745     ENDBLOCK
746     ENDBLOCK
747     ENDBLOCK
748     ENDBLOCK
749     ENDBLOCK
750     ENDBLOCK
751     ENDBLOCK
752     ENDBLOCK
753     ENDBLOCK
754     ENDBLOCK
755     ENDBLOCK
756     ENDBLOCK
757     ENDBLOCK
758     ENDBLOCK
759     ENDBLOCK
760     ENDBLOCK
761     ENDBLOCK
762     ENDBLOCK
763     ENDBLOCK
764     ENDBLOCK
765     ENDBLOCK
766     ENDBLOCK
767     ENDBLOCK
768     ENDBLOCK
769     ENDBLOCK
770     ENDBLOCK
771     ENDBLOCK
772     ENDBLOCK
773     ENDBLOCK
774     ENDBLOCK
775     ENDBLOCK
776     ENDBLOCK
777     ENDBLOCK
778     ENDBLOCK
779     ENDBLOCK
780     ENDBLOCK
781     ENDBLOCK
782     ENDBLOCK
783     ENDBLOCK
784     ENDBLOCK
785     ENDBLOCK
786     ENDBLOCK
787     ENDBLOCK
788     ENDBLOCK
789     ENDBLOCK
790     ENDBLOCK
791     ENDBLOCK
792     ENDBLOCK
793     ENDBLOCK
794     ENDBLOCK
795     ENDBLOCK
796     ENDBLOCK
797     ENDBLOCK
798     ENDBLOCK
799     ENDBLOCK
800     ENDBLOCK
801     ENDBLOCK
802     ENDBLOCK
803     ENDBLOCK
804     ENDBLOCK
805     ENDBLOCK
806     ENDBLOCK
807     ENDBLOCK
808     ENDBLOCK
809     ENDBLOCK
810     ENDBLOCK
811     ENDBLOCK
812     ENDBLOCK
813     ENDBLOCK
814     ENDBLOCK
815     ENDBLOCK
816     ENDBLOCK
817     ENDBLOCK
818     ENDBLOCK
819     ENDBLOCK
820     ENDBLOCK
821     ENDBLOCK
822     ENDBLOCK
823     ENDBLOCK
824     ENDBLOCK
825     ENDBLOCK
826     ENDBLOCK
827     ENDBLOCK
828     ENDBLOCK
829     ENDBLOCK
830     ENDBLOCK
831     ENDBLOCK
832     ENDBLOCK
833     ENDBLOCK
834     ENDBLOCK
835     ENDBLOCK
836     ENDBLOCK
837     ENDBLOCK
838     ENDBLOCK
839     ENDBLOCK
840     ENDBLOCK
841     ENDBLOCK
842     ENDBLOCK
843     ENDBLOCK
844     ENDBLOCK
845     ENDBLOCK
846     ENDBLOCK
847     ENDBLOCK
848     ENDBLOCK
849     ENDBLOCK
850     ENDBLOCK
851     ENDBLOCK
852     ENDBLOCK
853     ENDBLOCK
854     ENDBLOCK
855     ENDBLOCK
856     ENDBLOCK
857     ENDBLOCK
858     ENDBLOCK
859     ENDBLOCK
860     ENDBLOCK
861     ENDBLOCK
862     ENDBLOCK
863     ENDBLOCK
864     ENDBLOCK
865     ENDBLOCK
866     ENDBLOCK
867     ENDBLOCK
868     ENDBLOCK
869     ENDBLOCK
870     ENDBLOCK
871     ENDBLOCK
872     ENDBLOCK
873     ENDBLOCK
874     ENDBLOCK
875     ENDBLOCK
876     ENDBLOCK
877     ENDBLOCK
878     ENDBLOCK
879     ENDBLOCK
880     ENDBLOCK
881     ENDBLOCK
882     ENDBLOCK
883     ENDBLOCK
884     ENDBLOCK
885     ENDBLOCK
886     ENDBLOCK
887     ENDBLOCK
888     ENDBLOCK
889     ENDBLOCK
890     ENDBLOCK
891     ENDBLOCK
892     ENDBLOCK
893     ENDBLOCK
894     ENDBLOCK
895     ENDBLOCK
896     ENDBLOCK
897     ENDBLOCK
898     ENDBLOCK
899     ENDBLOCK
900     ENDBLOCK
901     ENDBLOCK
902     ENDBLOCK
903     ENDBLOCK
904     ENDBLOCK
905     ENDBLOCK
906     ENDBLOCK
907     ENDBLOCK
908     ENDBLOCK
909     ENDBLOCK
910     ENDBLOCK
911     ENDBLOCK
912     ENDBLOCK
913     ENDBLOCK
914     ENDBLOCK
915     ENDBLOCK
916     ENDBLOCK
917     ENDBLOCK
918     ENDBLOCK
919     ENDBLOCK
920     ENDBLOCK
921     ENDBLOCK
922     ENDBLOCK
923     ENDBLOCK
924     ENDBLOCK
925     ENDBLOCK
926     ENDBLOCK
927     ENDBLOCK
928     ENDBLOCK
929     ENDBLOCK
930     ENDBLOCK
931     ENDBLOCK
932     ENDBLOCK
933     ENDBLOCK
934     ENDBLOCK
935     ENDBLOCK
936     ENDBLOCK
937     ENDBLOCK
938     ENDBLOCK
939     ENDBLOCK
940     ENDBLOCK
941     ENDBLOCK
942     ENDBLOCK
943     ENDBLOCK
944     ENDBLOCK
945     ENDBLOCK
946     ENDBLOCK
947     ENDBLOCK
948     ENDBLOCK
949     ENDBLOCK
950     ENDBLOCK
951     ENDBLOCK
952     ENDBLOCK
953     ENDBLOCK
954     ENDBLOCK
955     ENDBLOCK
956     ENDBLOCK
957     ENDBLOCK
958     ENDBLOCK
959     ENDBLOCK
960     ENDBLOCK
961     ENDBLOCK
962     ENDBLOCK
963     ENDBLOCK
964     ENDBLOCK
965     ENDBLOCK
966     ENDBLOCK
967     ENDBLOCK
968     ENDBLOCK
969     ENDBLOCK
970     ENDBLOCK
971     ENDBLOCK
972     ENDBLOCK
973     ENDBLOCK
974     ENDBLOCK
975     ENDBLOCK
976     ENDBLOCK
977     ENDBLOCK
978     ENDBLOCK
979     ENDBLOCK
980     ENDBLOCK
981     ENDBLOCK
982     ENDBLOCK
983     ENDBLOCK
984     ENDBLOCK
985     ENDBLOCK
986     ENDBLOCK
987     ENDBLOCK
988     ENDBLOCK
989     ENDBLOCK
990     ENDBLOCK
991     ENDBLOCK
992     ENDBLOCK
993     ENDBLOCK
994     ENDBLOCK
995     ENDBLOCK
996     ENDBLOCK
997     ENDBLOCK
998     ENDBLOCK
999     ENDBLOCK
1000    ENDBLOCK

```

من المفيد دراسة أمثلة الماكرو تعريفات المذكورة في كتاب [Y.Tabourier ، أ. روشفلد Rochfeld ونس. فرانك C. Frank] . إنها عبارة عن ماكرو تعريفات تسمح ببناء برنامج مؤول بصورة بنوية مركبة . والكتاب يعرض للماكرو WHILE زائد شرط ، DO ، ENDWHILE ، IF ، THEN ، ELSE ، ENDIF ، BLOCK ، ENDBLOCK ، بالتفصيل ، وتقوم هي باستدعاء 2 ماكرو تديران مكسداً من المؤثرات .

23 . نصائح في البرمجة

ليس هدفنا عرض طريقة في البرمجة تشبه البرمجة الإنشائية ، ولكن ببساطة إعطاء بعض النصائح الناتجة عن الخبرة العملية لمختلف الطرق . هذه الملاحظات يمكن أن توسّع لتشمل جميع أنواع المؤول وفي بعض الأحيان تنطبق على اللغات المتطورة .

1.23 . تركيبة المعالجة

1.1.23 . البرمجة الزجالية

هي عبارة عن قاعدة عامة في البرمجة . هناك قائمة من تقسيم المسألة إلى زُجل (أقسام) صغيرة قدر الإمكان . كل زجلة تحلُّ مهمة معينة والبرنامج الرئيسي يؤمّن ترابط الأقسام فيما بينها . ولقد عرضنا في الفصلين 20 و 21 . طريقة استعمال وسائل التقطيع وإنشاء البرامج - الثانية .

2.1.23 . تقديم وإعداد

البرنامج بلغة المؤول هو عادة عبارة عن نص غير واضح ، ويحد المصنّم صعوبة في تعديل وإعادة قراءة ما كتبه منذ اللحظة التي يترك فيها برنامجه جانباً لبعض الوقت . يجب إذا كتابة الملاحظات بعد كل تعليمة لتوضيح نص البرنامج . الأوامر SPACE N (إدخال عدة أسطر n بيضاء) ، EJECT (عبور إلى الصفحة التالية) و PRINT NOGEN (إلغاء توليد كود الماكرو تعليقات) تسمح بتسهيل نص البرنامج بجعله أكثر وضوحاً .

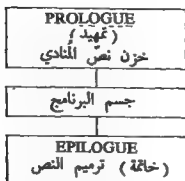
البرنامج المؤرّد بملاحظات يبدأ بتحديد مهمة الزجلة ، وروابطها مع الزجل الباقية كما يحتوي على أسماء ومهمة التحويلات والمراصف المستعملة .

2.23 . تركيبة الزجلة

1.2.23 . التمهيد والخاتمة (Prologue و epilogue)

بإمكاننا اعتبار كل برنامج وكأنه برنامج ثانوي لبرنامج آخر . الزجلة الرئيسية هي عبارة عن برنامج ثانوي من نظام التشغيل ويجب عليها أن تحفّز نتائج البرنامج النهائي .

- تقنية الحزن وترميم نص النادي هي أساسية وقد جرى تعريفها في الفصل 21 . بإمكاننا إنشاء كل زجلة على الشكل التالي :



هناك فائدة للمبرمج في تحقيق التمهيدات والخاتمة الخاصة به حسب القواعد المتفق عليها والمذكورة في الفصل 21 . الاتصال بين الزجل المكتوبة في اللغات المختلفة سيكون مبسطاً وأكثر من خطأ سيتم تفاديه باستعمال مناسب للمراسف . لقد ذكرنا مثلاً في الفصل 22 الماكرو - تعليمة PROLOGUE التي تحل هذه المسألة وتوفر على المبرمج كتابة صعبة للتعليقات الأولية .

2.2.23 . جسم البرنامج

يتألف من تعليقات قابلة للتنفيذ ومن معطيات . سنضع المعطيات بعد التعليقات . استعمال الأمر LTORG سيسمح لنا بوضع تأويل الثوابت من نوع حرفي في المكان الذي نرغب فيه . المتحولات والثوابت ستكون إذاً مترابطة ، مما يجعلها متجاورة في كل dump وستسمح بإجراء تقسيم سهل إلى أقسام إذا كنا نرغب بجعل البرنامج مبسطاً للتعديل والاختبار . سنستعمل عند الحاجة أوامر حجز مكان من الذاكرة بواسطة DC أكثر من بواسطة DS معنيين بهذه الطريقة منطقة من الذاكرة بقيمة سوف يمكننا مراقبتها في dump (دلق) .

إستعمال - المرجعيات الرمزية

إن كتابة LR1,2 تعود عملياً إلى العمل بلغة الآلة . وفي المقابل فإن كتابة LR1,R2 بعد تعريف الرمز R1 و R2 بواسطة EQU معناها إستعمال إمكانيات ومرونة الترميز ، والمرجمان R1 و R2 يظهران في جدول الرموز . من الأفضل أيضاً إعطاء المراسف والمتحولات أسماء مؤكدة حرفياً كما جرى في أمثلة الفصل 15 . فليس من المزعج أكثر من قراءة التعليقات التي تذكر المراسف بشكل ظاهر .

هكذا ، فكتابة B * + 14 تؤدي إلى سيطرة تكمن في تجميد البرنامج ، ويصبح من

غير الممكن إدخال تعليقات جديدة بين العنوانين * و14+ * دون تعديل تعليقات الترميز . لذا فمن الأفضل تعريف وسم ALPHA وكتابة B ALPHA . الكتابات من النوع n + * لا يجب أن تُستعمل إلا داخل الماكرو - تعريفات . وختاماً يجب على البرنامج أن يكون دائماً مكتوباً مع أخذ التعديلات اللاحقة بعين الاعتبار إضافة إلى مسائل الصيانة .

هكذا يجب تعريف جميع العناصر القابلة للتعديل في البرنامج بواسطة EQU . هذا الأمر هو شديد الأهمية . وفي حالة التعديل فهو يسمح بتخفيض عدد التغيرات المطلوب إجراؤها . ويقدم فائدة تكمن في جعل التعليقات « مزودة بملاحظات » . إن التمرين 8.13 يوضح لنا ذلك .

الخاصية - طول

تسمح بجعل البرنامج يحتوي على متغيرات . كل تعديل على طول المنطقة لن يؤثر على التعليقات التي تذكر هذا الطول بواسطة LZONE .

تركيبة منطقة المخططات

بدلاً من مراجعة أقسام (field) نفس المنطقة بواسطة المسافة بالنسبة لبداية المنطقة ، من الأفضل تخصيص (بواسطة EQU) أسماء رمزية لمختلف هذه الأقسام . كل تعديل على التركيبة يصبح عندئذ سهلاً . يُوضح لنا التمرين 2.8 تعريف تركيبة كهذه .

إستعمال الكود الحرفي

يترك للمؤول مهمة تعريف الثوابت الضرورية دون إسهاب . هذه الثوابت يمكن أن تكون مجموعة في المكان المطلوب بواسطة الأمر LTORG .

كتابة الأوسمة

سنُعرف الأوسمة بواسطة الأمر DSOH . نتأكد من تسطير (إصطفاة) حدود نصف - كلمة والوسم لن يعود مرتبطاً بالتعليمة الموجودة في الجهة المقابلة له . سيصبح ممكناً عكس بعض التعليقات بواسطة معالجة بسيطة للبطاقات .

إستعمال المراسف

قبل أية عملية برمجة يجب التنقيب عن الخيارات التي يقوم بها النظام لاستعمال المراسف . وقد جرى عرض ذلك في الفصل 21 . وللمبرمج فائدة في إجراء نفس الاختيار لأسباب تتعلق بالتوافق . فلنذكر أن OS :

يشحن في R15 عنوان نقطة الدخول ،

في R14 عنوان العودة ،

في R13 عنوان المنطقة SAVE AREA .

ويستعمل R0 لارسال نتيجة مهمة من نوع (FUNCTION في فورتران) ،
و R1 لإرسال عنوان لائحة متغيرات إلى برنامج ثانوي .

بعض التعليقات (TRT, EMDK) تستعمل المرصفين R1 و R2 . سيختار المبرمج
مراسيف القاعدة من ضمن المراسيف 12 ، 11 و... ومراسيف العمل من ضمن
المراسيف 3 ، 4 ، ...

إستعمال الماكرو - لغة (MACRO-language)

باستعمال الماكرو لغة فإن المؤلف يقترب من اللغة المتطورة . وهي تسمح للمبرمج
بأن يكون مزوداً بوسائل إعداد البرنامج وجعله إنشائياً (مركباً) . وسيكون بإمكانه ،
مثلاً ، إنشاء ماكرو - تعريف يسمح له بمتابعة أثر البرنامج عند التنفيذ بواسطة طباعة
الأوسمة خلال مرحلة الاختبار . عند التأويل النهائي فإن توليد الماكرو - تعليمة سيتم
إلغاؤه بواسطة تعديل بسيط لقيمة متحولة التأويل . ولن تولد أوسمة بواسطة
ETIQ DS OH . بإمكان المبرمج أن يقوم أيضاً بإنشاء ماكروتعريفات تولد مثلاً
تعليقات من نوع WHILE ، DO ، ENDDO ... وإمكان الأوسمة أن تختفي من
النص المطلوب تأويله ويصبح البرنامج أكثر إنشائياً .

وفي النهاية فإن الزجالة يمكن أن تحصل على التركيبة التالية :

MACRO-DEFINITIONS
COMMENTAIRES
EQU ...
PROLOGUE
CORPS
EPILOGUE
ZONE DE DONNÉES

ماكرو تعليمات
ملاحظات
EQU...
مقدمة
خاتمة
جسم البرنامج
منطقة المعطيات
3.23 . الخلاصة

بشكل عام لا تؤيد المبالغة في استعمال الحيل والحلق من قبل المبرمج . فالبرنامج
« المحايل » هو غائص على العموم بالنسبة للقارئ المبتدئ ، وأحياناً تقترب الحيل من
الإضهار المبهم ويمكننا هنا تصور للمشاكل التي قد تعترض عمل فريق صيانة البرامج .
في لغة المؤلف تختلف المسألة نوعاً ما . فبالإمكان إقامة عدد معين من الحيل
ضمن نطاق تقنيات الحل وفي هذا الإطار يتعين على المبرمج أن يعرفها . لقد ذكرنا
خلال الأمثلة والتأهين عدداً كبيراً من الصفات المنتشرة كغاية بشكل يسمح لنا باعتبارها
كأدوات أساسية . هذا هو السبب الذي يجعلنا نصرّح ، دراستها من قبل القارئ بعناية
واهتمام .

حلول التمارين

التمارين 1.2 -	النظام 16	النظام 2	النظام 10
	F	1111	15
	23	10 0011	35
	100	1 0000 0000	256
	400	100 0000 0000	1024
	15C.8	1 0101 1100.1	348.5

التمارين 2.2 -	النظام 2	النظام 10	النظام 16
	11 1010	58	3A
	1111 1111 1111	4095	FFF (~1000-1)
	1 1010 0011 1011	6715	1A3B
	1010 1011 1100	2748	ABC

التمارين 3.2 - المكمل إلى FFFF : E5C4

المكمل إلى E5C5:2

الطرح بواسطة جمع المكمل إلى 2 (نتحقق ما إذا كان يجب لنا تجاهل المرحّل) النتيجة : 1081 .

1A3B على 32 بتة : 3B 1A 00 00

E5C5 على 32 بتة : C5 E5 FF FF

التمارين 4.2 - تكويد الإشارة والقيمة المطلقة : $15.16^3 + 16^6 + 4.16^7 -$

التكويد بالمكمل إلى 2 : $16^3 + 14.16^6 + 3.16^7 -$

التكويد بالفاصلة المتحركة : $(15.16^{-3}) - 16^4$

العكس (الضد) بالإشارة والقيمة المطلقة : 41 F0 00 00

العكس بالمكمل إلى 2 : 3E 10 00 00

العكس بالفاصلة المتحركة : 41 F0 00 00 (معاير)

لا يمكن لهذا التمثيل أن يكون تمثيل عدد مكوّد بالنظام DCB (عشري مكوّد ثنائيًا) .

$$\begin{aligned}
 C5\ 03\ 20\ 00 &= -16^5(3.16^{-2}+2.16^{-3}) & \text{تمرین 5.2} \\
 &= -\frac{16^5}{16} \cdot 16(3.16^{-2}+2.16^{-3}) \\
 &= -16^4(3.16^{-1}+2.16^{-2}) = C4\ 32\ 00\ 00
 \end{aligned}$$

TAB DC 100AL1(*-TAB+1) تمرین 1.8
 TAB DC 100A((*-TAB)/4+1)

MOSS DS 0CL13 L'MOSS = 13
 SEXE DS CL1 L'SEXE = 1
 DATE DS 0CL4 L'DATE = 4
 ANNEE DS CL2 L'ANNEE = 2
 MOIS DS CL2 L'MOIS = 2
 LIEUNAT DS 0CL5 L'LIEUNAT = 5
 DEPART DS CL2 L'DEPART = 2
 COM DS CL3 L'COM = 3
 NO DS CL3 L'NO = 3

تمرین 2.8 -

Z1 DS 0F تأطير على حد كلمة
 PRIX DS 0CL12 ZL8
 QTE DS ZL4
 ORG Z1
 Z2 DS 0CL14
 NO DS F
 TEXTE DS CL10

تمرین 3.8 -

تمرین 1.9 -

LJC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000000				1	CSECT
000000	5810 C92C	0002C	00000	2	USING a,12
000000	5833 C02C	0002C		3	L 8D
000000	0000			4	L 3D(3)
	*** ERROR ***			5	A,D
000000	0000 0300	00004		6	ST D,X'4'(3,C)
	*** ERROR ***			7	
000000	5803 0000	00000		8	L A,B'1011'(3)
000012	0000 0000	00040			D,E(B)
	*** ERROR ***			9	
000016	5801 C040	00040			L A,E(B)
00001A	0200 A000 C92C	00000	0002C	10	MVC A(B,C)+D
000020	0203 C040 C02C	00040	0002C	11	MVC E(L'D)+D
000028	5820 C030	00030		12	L 2,D+L'D
		00000		13	EQU 0
		00001		14	EQU 1
		0000A		15	EQU 10
00002C				16	DS 8F
000040				17	DS 12F
				18	END

ASSEMBLER DIAGNOSTICS AND STATISTICS

97M1 ERROR CODE MESSAGE

5	IF0217	RELOCATABILITY ERROR NEAR OPERAND COLUMN 4
6	IF0217	RELOCATABILITY ERROR NEAR OPERAND COLUMN 2
8	IF0217	RELOCATABILITY ERROR NEAR OPERAND COLUMN 2

NUMBER OF STATEMENTS FLAGGED IN THIS ASSEMBLY = 3
 HIGHEST SEVERITY WAS 12

تمرين 1.11 -

حلول أخرى بواسطة «أو المختصرة» أو «الإزاحات» .

LA R,0
SR R,R

تمرين 2.11 -

LCR R,R

تمرين 3.11 -

LN R,=H'-1'

نستعمل كون نصف الكلمة موسماً إلى كلمة قبل العملية بواسطة انتشار
بته ذات وزن قوي .

تمرين 4.11 -

LA R,2048
LA R,4095
L R,=F'4095' : لو { LA R,4095'
LA R,1(0,R)

تمرين 5.11 -

LA R,4(0,R)

تمرين 6.11 -

MV1 ZONE,C'0'
MVC ZONE+1(L'ZONE-1),ZONE

نستعمل كون الحركة تتم بآلية بعد بآلية من اليسار إلى اليمين .

تمرين 1.12 -

N DC ... عدد التكرارات .
R1 EQU 3

L R1,N
TRAIT ---

BCT R1,TRAIT

• معالجة التكرار

تمرين 2.12 -

تسمح الماكرو تعليمية SNAP بالحصول على عمليات دلق («dumps»)
جزئية في الذاكرة . ويجب أن تسبقها ماكرو OPEN (فتح سجل) . في
حالتنا الحاضرة يمتد الدلق dump من العنوان SNAPDEB حتى العنوان
SNAPFIN . تعطي الكلمة PSW عنوان بداية SNAP . وتعطي الجهة

اليعني من dump ، حتى يكون ذلك ممكناً ، تفسير محتوى الذاكرة الثنائي على شكل سمات . وسيتمرن القارئ بمحاولة إيجاد محتويات مختلف مناطق البرنامج عبر حساب العناوين من خلال العنوان الاسامي الموجود في المرفص 12 .

(انظر اللائحة listing في الصفحة اللاحقة) .

تمرين 3.12 -

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
000000				1	DEBUT	START 0
				2		PRINT NOGEA,DATA
				3		عكس سلسلة من السمات *
			000003	4	WORK	EQV 3
			000004	5	IND1	EQU 4
			000005	6	INC2	EQU 5
000000				7	SHAPDES	DS 2H
000000				8	PROLOGUF	DS 0H
000000	90EC 009C	0C00C		9	STM	10,12,12(13)
000004	10CF		00000C	10	USING	DEBUT,12
000006	5000 C07A	0C078		11	LR	12,15
00000A	4100 C07A	0C074		12	ST	13,SAVE+4
				13	LA	13,SAVE
00000E	4150 C000	000000		14	LA	INC2,0
000012	4140 C005	000005		15	LA	INC1,L*CH1
000016				16	CS	0H
000016	433A C067	000067		17	BCL	
00001A	4235 C060	000060		18	IC	WORK,CH1-1(IND1)
00001E	4155 00C1	0000C1		19	STC	WORK,CH2(IND2)
000022	4640 C016	00016		20	LA	INC2,1(IND2)
				21	BCF	INC1,BCL

تمرين 4.12 - BCTR R,0

تمرين 5.12 - BALR R,0

تمرين 1.13 - XC ZONE,ZONE منطقة بطول L
XR R1,R1 مرفص
NI OCTET,X'00' بايت

تمرين 2.13 - لنفترض التعليمة في العنوان INSTR . إذن يوجد كود الطول في INSTR + 1 (تعليمة بنسق SS) .

#1 (إعادة تصفير)

01 (حيث XX هو الطول ناقص واحد)

علينا أن نتذكر أنه ، بالنسبة للتعليقات من النوع SS ، الطول المؤول هو الطول ناقص واحد .

تمرين 3.13 - XC ZONE1(L),ZONE2
XC ZONE2(L),ZONE1
XC ZONE1(L),ZONE2

XR R1,R2
XR R2,R1
XR R1,R2

PACK OCTET,OCTET أو UNPK OCTET,OCTET

تمرين 4.13 - كما في التمرين 6.11 ، نستعمل كون العمليات مع التعليمات MVC،
CLC ... تجري بايتة بعد بايتة مع انتشار من اليسار إلى اليمين .

```

CLI    ZONE,X'00'  (مقارنة أول بايتة مع 0)
BNE    NONZERO
CLC    ZONE+1(L'ZONE-1),ZONE
BNE    NONZERO
BE     ZERO

```

التعرف إلى « الفراغ (blank) » يتم عبر للمقارنة مع X'40' .

تمرين 5.13 -

```

XI      ++5,X'F0'
NOP     ETIQ

```

تمرين 6.13 -

```

NOP     ETIQ
XI      +-3,X'F0'

```

تمرين 7.13 . REF و INCR يشكّان مرصفاً مزدوجاً يحتوي الزيادة والمرجع بالنسبة
للتعليمة BXLE و NOMBRE هو عنوان العدد. أما PTR فهو مرصف
مصوّب (مؤشّر) .

			NOMBRE					
			↑					
			F0	F0	F4	F2	F1	
			↑PTR			↑REF		
TEST	LA	PTR,NOMBRE						
	LA	REF,L'NOMBRE-1(PTR)						
	LA	INCR,1						
	CLI	0(PTR),C'0'						
	BNE	SUITE						
	MVI	0(PTR),C'1'						
SUITE	BXLE	PTR,INCR,TEST						
						

تمرين 8.13 -

```

INDIC   DC    X'00'
INDEC   EQU   X'80'
INDECR  EQU   X'40'
INMAIT  EQU   X'20'
-----
OI      INDIC,INMAIT
OI      INDIC,INDEC+INMAIT
NI      INDIC,X'FF'- (INDEC+INDECR)
-----
TW      INDIC,INMAIT
HO      ALPHA
-----
TW      INDIC,INDEC+INMAIT
BO      BETA
BN      GAMMA
BZ      DELTA

```

مع هذا الحل فإنّ التعديل للتعليق بِـ INDEC يُترجم بواسطة :

INDLEC EQU X'01'

لا تتأثر أي تعليمة تحليل موضوع أو اختبار . والأمر لا يكون كذلك إن نحن لم نستعمل EQU لتحديد المؤشرات الثنائية ، فحينئذٍ لكان الكود مجمداً بسبب ظهور القيم X'80' ... في قلب التعليمات نفسها .
من جهة أخرى فإن هذه التقنية تحوّل التعليمات لأن تصبح مؤقّدة ذاتياً .

تمرين 1.14 -
SLL R,32
SRL R,32 أو

تمرين 2.14 - يكتب الضرب بـ 2^3 : R,3 SLA
القسمة على 16 : R,4 SRA

ترافق القسمة عملية بتر (قطع) . والتمثيل بالمكمل إلى 2 يعادل $15/2 +$
تعطي 7 في حين أنّ $15/2 -$ تعطي -8 .

تمرين 3.14 - R SLDA R,0
ZERO
هو مرصف مزدوج

تمرين 4.14 . أثناء عملية إزاحة دائرية إلى اليسار نحاول إعادة إدخال كلّ بته خارجة في جهة اليمين . العمل يتم على مرصف مزدوج . بعد تصفير مرصف اليسار نجري إزاحة مزدوجة بشكل يسمح بأن نجد من جديد في مرصف اليسار البتات المفقودة في مرصف اليمين . ونتج لنا تعليمة أو (OR) بإعادة وضعها في مرصف اليمين . هنا نجري إزاحة دائرية من أربعة مواقع على المرصف 7 .

SLL 6,32 تصفير
SLIL 6,4
OR 7,6

تمرين 1.18 - نستعمل تعليمة TR « بالقلب » .

TR	CLE,ARTICLE	
---	----	
ARTICLE	DC CL3D'ABCDEFGH1J'	(قراءة)
CLE	DS 0CL5	(مفتاح)
DC	HL'5,6,7,1,2'	

تمارين 2.18 -

نستعمل التعليمة TR

TR CHAINE (B),TABLE

DC C'0123456789ABCDEF' (جدول)

DC 2F (سلسلة)

ملحقات

جدول تكوين السيات
جدول أبجدي للتعليقات
أوامر المؤول
مميزات الثوابت
كود حرفي (تلكيري) موسّع

جدول تكرید السیات

بطاقة مقبولة	رمز مطبوعة	تكريري مقبولة	سلسل مقبولة	عشري	بطاقة مقبولة	رمز مطبوعة	تكريري مقبولة	سلسل مقبولة	عشري
0	00		12-0-0-1	64	40	STH	لا	40	64
1	01		12-0-1	65	41	LA	تكريري	41	65
2	02		12-0-2	66	42	STC	يائس	42	66
3	03		12-0-3	67	43	IC		43	67
4	04	SPM	12-0-4	68	44	EX		44	68
5	05	BALR	12-0-5	69	45	BAL		45	69
6	06	BCTR	12-0-6	70	46	BCT		46	70
7	07	BCR	12-0-7	71	47	BC		47	71
8	08	SSK	12-0-8	72	48	LJI		48	72
9	09	ISK	12-0-9	73	49	CH		49	73
10	0A	SVC	12-0-10	74	4A	AR		4A	74
11	0B		12-0-11	75	4B	SH	(نقطة)	4B	75
12	0C		12-0-12	76	4C	MH	()	4C	76
13	0D		12-0-13	77	4D			4D	77
14	0E	MVCL	12-0-14	78	4E	CVD	+	4E	78
15	0F	CLCL	12-0-15	79	4F	CVB		4F	79
16	10	LPR	12-11-0-1	80	50	ST	&	50	80
17	11	LNR	11-0-1	81	51			51	81
18	12	LTR	11-0-2	82	52			52	82
19	13	LGR	11-0-3	83	53			53	83
20	14	NE	11-0-4	84	54	N		54	84
21	15	CLR	11-0-5	85	55	CL		55	85
22	16	OL	11-0-6	86	56	O		56	86
23	17	XR	11-0-7	87	57	X		57	87
24	18	LR	11-0-8	88	58	L		58	88
25	19	CR	11-0-9	89	59	C		59	89
26	1A	AR	11-0-10	90	5A	A		5A	90
27	1B	SR	11-0-11	91	5B	S		5B	91
28	1C	MR	11-0-12	92	5C	M		5C	92
29	1D	DR	11-0-13	93	5D	D		5D	93
30	1E	ALR	11-0-14	94	5E	AL		5E	94
31	1F	SLR	11-0-15	95	5F	SL		5F	95
32	20	LPDR	11-0-16	96	60	STD	-	60	96
33	21	LNDR	0-0-1	97	61		/	61	97
34	22	LTDR	0-0-2	98	62			62	98
35	23	LCDR	0-0-3	99	63			63	99
36	24	HDR	0-0-4	100	64			64	100
37	25	LRDR	0-0-5	101	65			65	101
38	26	MXR	0-0-6	102	66			66	102
39	27	MXDR	0-0-7	103	67	MXD		67	103
40	28	LDR	0-0-8	104	68	LD		68	104
41	29	CDR	0-0-9	105	69	CD		69	105
42	2A	ADR	0-0-10	106	6A	AD		6A	106
43	2B	SDR	0-0-11	107	6B	SD		6B	107
44	2C	MDR	0-0-12	108	6C	MD	%	6C	108
45	2D	DDR	0-0-13	109	6D	DD	-	6D	109
46	2E	AWR	0-0-14	110	6E	AW	?	6E	110
47	2F	SWR	0-0-15	111	6F	SW		6F	111
48	30	LPER	12-11-0-0-1	112	70	STE		70	112
49	31	LNER	9-1	113	71			71	113
50	32	LTER	9-2	114	72			72	114
51	33	LGER	9-3	115	73			73	115
52	34	HER	9-4	116	74			74	116
53	35	LHER	9-5	117	75			75	117
54	36	AXR	9-6	118	76			76	118
55	37	EXR	9-7	119	77			77	119
56	38	LER	9-8	120	78	LE		78	120
57	39	CEB	9-9	121	79	CE		79	121
58	3A	AER	9-10	122	7A	AE		7A	122
59	3B	SEB	9-11	123	7B	SE	*	7B	123
60	3C	MER	9-12	124	7C	ME		7C	124
61	3D	DER	9-13	125	7D	DE		7D	125
62	3E	AUR	9-14	126	7E	AU		7E	126
63	3F	SUR	9-15	127	7F	SU	=	7F	127

جدول تكويد السهات

بطاقة مكتوبة	بسمية مطبوعة	حرفي تذكيري	ساحس عشري	١-٩	بطاقة	بسمية مطبوعة	حرفي تذكيري	ساحس عشري	بطاقة مكتوبة
128	80	SSM		192	C0			120	12-0-8-1
129	81			193	C1			121	12-0-8-2
130	82	LPSW		194	C2		A	122	12-0-8-3
131	83			195	C3		B	123	12-0-8-4
132	84	WRD		196	C4		C	124	12-0-8-5
133	85	RDD		197	C5		D	125	12-0-8-6
134	86	ROH		198	C6		E	126	12-0-8-7
135	87	ROLE		199	C7		F	127	12-0-8-8
136	88	SRL		200	C8		G	128	12-0-8-9
137	89	SRL		201	C9		H	129	12-0-9-0
138	9A	SRL		202	CA		I	130	12-0-9-1
139	9B	SRL		203	CB			131	12-0-9-2
140	9C	SRDL		204	CC			132	12-0-9-3
141	9D	SIDL		205	CD			133	12-0-9-4
142	9E	SRDA		206	CE			134	12-0-9-5
143	9F	SILDA		207	CF			135	12-0-9-6
144	9G	STM		208	D0			136	12-0-9-7
145	9H	TM		209	D1	MVN	J	137	12-0-9-8
146	9I	MVI		210	D2	MVC	K	138	12-0-9-9
147	9J	TS		211	D3	MOVZ	L	139	12-1-0-0
148	9K	NI		212	D4	NC	M	140	12-1-0-1
149	9L	CLJ		213	D5	CLC	N	141	12-1-0-2
150	9M	OI		214	D6	OC	O	142	12-1-0-3
151	9N	XI		215	D7	XC	P	143	12-1-0-4
152	9O	LM		216	D8		Q	144	12-1-0-5
153	9P			217	D9		R	145	12-1-0-6
154	9A			218	DA			146	12-1-0-7
155	9B			219	DB			147	12-1-0-8
156	9C	SRO		220	DC	TR		148	12-1-0-9
157	9D	TRO		221	DD	TRT		149	12-1-1-0
158	9E	HRO		222	DE	RD		150	12-1-1-1
159	9F	TCH		223	DF	EDMK		151	12-1-1-2
160	A0			224	ED			152	12-1-1-3
161	A1			225	E0			153	12-1-1-4
162	A2			226	E1			154	12-1-1-5
163	A3			227	E2		S	155	12-1-1-6
164	A4			228	E3		T	156	12-1-1-7
165	A5			229	E4		U	157	12-1-1-8
166	A6			230	E5		V	158	12-1-1-9
167	A7			231	E6		W	159	12-1-2-0
168	A8			232	E7		X	160	12-1-2-1
169	A9			233	E8		Y	161	12-1-2-2
170	AA			234	EA		Z	162	12-1-2-3
171	AB			235	EB			163	12-1-2-4
172	AC	STNSM		236	EC			164	12-1-2-5
173	AD	STORM		237	ED			165	12-1-2-6
174	AE	SIOP		238	EE			166	12-1-2-7
175	AF	MC		239	EF			167	12-1-2-8
176	B0			240	FO	SRP	0	168	12-1-2-9
177	B1	LRA		241	F1	MVO	1	169	12-1-3-0
178	B2			242	F2	PACK	2	170	12-1-3-1
179	B3			243	F3	UNPK	3	171	12-1-3-2
180	B4			244	F4		4	172	12-1-3-3
181	B5			245	F5		5	173	12-1-3-4
182	B6	STCTL		246	F6		6	174	12-1-3-5
183	B7	LCCTL		247	F7		7	175	12-1-3-6
184	B8			248	F8	ZAP	8	176	12-1-3-7
185	B9			249	F9	CP	9	177	12-1-3-8
186	BA	CS		250	FA	AP		178	12-1-3-9
187	BB	CDS		251	FB	SP		179	12-1-4-0
188	BC			252	FC	MP		180	12-1-4-1
189	BD			253	FD	DP		181	12-1-4-2
190	BE	CLM		254	FE			182	12-1-4-3
191	BF	ICM		255	FF			183	12-1-4-4

جدول أبجدي للتعليمات

النسق	منطقة العوامل	Format	منطقة العوامل
RR	R_1, R_2	SI	$D_1(B_1), I_2$
RR-M	M_1, R_2		
RR-1	R_1	S	$D_2(B_2)$
RR-I	I_1		
RX	$R_1, D_2(X_2, B_2)$	SS-1	$D_1(L, B_1), D_2(B_2)$
RX-M	$M_1, D_2(X_2, B_2)$	SS-2	$D_1(L_1, B_1), D_2(L_2, B_2)$
		SS-3	$D_1(L_1, B_1), D_2(B_2), I_3$
RS	$R_1, R_3, D_2(B_2)$	مراسف X, إزاحة D مفتاح بأربعة بتات قيمة ثنائية I طول L	
RS-M	$R_1, M_3, D_2(B_2)$		

الدالة (الوظيفة)	نسخة تدويري	COP ساعات عشري	النسق	موقع CC
Add	AR	1A	RR	*
Add	A	5A	RX	*
Add Decimal	AP	FA	SS-2	*
Add Halfword	AH	4A	RX	*
Add Logical	ALR	1E	RR	*
Add Logical	AL	5E	RX	*
AND	NR	14	RR	*
AND	N	54	RX	*
AND	NI	94	SI	*
AND	NC	D4	SS-1	*
Branch and Link	BALR	05	RR	
Branch and Link	BAL	45	RX	
Branch on Condition	BCR	07	RR-M	
Branch on Condition	BC	47	RX-M	
Branch on Count	BCTR	06	RR	
Branch on Count	BCT	46	RX	
Branch on Index High	BXHI	86	RS	
Branch on Index Low or Equal	BXLE	B7	RS	
Compare	CR	19	RR	*
Compare	C	59	RX	*
Compare and Swap	CS	BA	RS	*
Compare Decimal	CP	F9	SS-2	*
Compare Double and Swap	CDS	BB	RS	*
Compare Halfword	CH	49	RX	*
Compare Logical	CLR	15	RR	*
Compare Logical	CL	55	RX	*
Compare Logical	CLC	D5	SS-1	*
Compare Logical	CLJ	95	SI	*
Compare Logical Characters under Mask	CLM	BD	RS-M	*
Compare Logical Long	CLCL	0F	RR	*
Convert to Binary	CVB	4F	RX	
Convert to Decimal	CVD	4E	RX	

الدالة (الوظيفة) Fonction	حرفي تذكيري Mnemo- nique	ساحس عشري hexa- derimal	النسق Format	يحدد موضع CC
Divide	DR	1F	RR	
Divide	D	5D	RX	
Divide Decimal	DP	FD	SS-2	
Edit	ED	DE	SS-1	*
Edit and Mark	EDMK	DF	SS-1	*
Exclusive OR	XR	17	RR	*
Exclusive OR	X	57	RX	*
Exclusive OR	XI	97	SI	*
Exclusive OR	XC	D7	SS-1	*
Execute	EX	44	RX	
Insert Character	IC	43	RX	
Insert Characters under Mask	ICM	BF	RS-M	*
Load	LR	18	RR	
Load	L	58	RX	
Load Address	LA	41	RX	
Load and Test	LTR	12	RR	*
Load Complement	LCR	13	RR	*
Load Halfword	LH	48	RX	
Load Multiple	LM	98	RS	
Load Negative	LNR	11	RR	*
Load Positive	LPR	10	RR	*
Monitor Call	MC	AF	SI	
Move	MVI	92	SI	
Move	MVC	D2	SS-1	
Move Long	MVCL	0E	RR	*
Move Numerics	MVN	D1	SS-1	
Move with Offset	MVO	F1	SS-2	
Move Zones	MVZ	D3	SS-1	
Multiply	MR	1C	RR	
Multiply	M	5C	RX	
Multiply Decimal	MP	FC	SS-2	
Multiply Halfword	MH	4C	RX	
OR	OR	16	RR	*
OR	O	56	RX	*
OR	OI	96	SI	*
OR	OC	D6	SS-1	*
Pack	PACK	F2	SS-2	
Set Program Mask	SPM	04	RR-1	
Shift and Round Decimal	SRP	F0	SS-3	*
Shift Left Double	SLDA	8F	RS	*
Shift Left Double Logical	SLDL	8D	RS	
Shift Left Single	SLA	8B	RS	*
Shift Left Single Logical	SLL	89	RS	
Shift Right Double	SRDA	8E	RS	*
Shift Right Double Logical	SRDL	8C	RS	
Shift Right Single	SRA	8A	RS	*
Shift Right Single Logical	SRL	88	RS	
Store	ST	50	RX	
Store Character	STC	42	RX	
Store Characters under Mask	STCM	BE	RS-M	
Store Clock	STCK	B205	S	*
Store Halfword	STH	40	RX	
Store Multiple	STM	90	RS	

Subtract	SR	1B	RR	*
Subtract	S	5B	RX	*
Subtract Decimal	SP	FB	SS-2	*
Subtract Halfword	SH	4B	RX	*
Subtract Logical	SLR	1F	RR	*
Subtract Logical	SL	5F	RX	*
Supervisor Call	SVC	0A	RR-I	
Test and Set	TS	93	S	*
Test under Mask	TM	91	SI	*
Translate	TR	DC	SS-1	
Translate and Test	TRO	DD	SS-1	*
Unpack	UNPK	F3	SS-2	
Zero and Add Decimal	ZAP	FB	SS-2	*

تعليمات حسابية بالفاصلة المتحركة

Add Normalized, Extended	AXR	36	RR	*
Add Normalized, Long	ADR	2A	RR	*
Add Normalized, Long	AD	6A	RX	*
Add Normalized, Short	AER	3A	RR	*
Add Normalized, Short	AE	7A	RX	*
Add Unnormalized, Long	AWR	2E	RR	*
Add Unnormalized, Long	AW	6E	RX	*
Add Unnormalized, Short	AUR	3E	RR	*
Add Unnormalized, Short	AU	7E	RX	*
Compare, Long	CDR	29	RR	*
Compare, Long	CD	69	RX	*
Compare, Short	CER	39	RR	*
Compare, Short	CE	79	RX	*
Divide, Long	DDR	2D	RR	
Divide, Long	DD	6D	RX	
Divide, Short	DER	3D	RR	
Divide, Short	DE	7D	RX	
Halve, Long	HDR	24	RR	
Halve, Short	HE	34	RR	
Load and Test, Long	LTDR	22	RR	*
Load and Test, Short	LTER	32	RR	*
Load Complement, Long	LCOR	23	RR	*
Load Complement, Short	LCER	33	RR	*
Load, Long	LDR	28	RR	
Load, Long	LD	68	RX	
Load Negative, Long	LWDR	21	RR	*
Load Negative, Short	LWER	31	RR	*
Load Positive, Long	LPDR	20	RR	*
Load Positive, Short	LPER	30	RR	*
Load Rounded, Extended Long	LRDR	25	RR	
Load Rounded, Long to Short	LRER	35	RR	
Load, Short	LER	38	RR	
Load, Short	LE	78	RX	
Multiply, Extended	MXR	26	RR	
Multiply, Long	MOR	2C	RR	
Multiply, Long	MO	6C	RX	
Multiply, Long/Extended	MXDR	27	RR	
Multiply, Long/Extended	MXD	67	RX	
Multiply, Short	MER	3C	RR	
Multiply, Short	ME	7C	RX	
Store, Long	STD	60	RX	

Store, Short	STE	70	RX	*
Subtract Normalized, Extended	SXR	37	RR	*
Subtract Normalized, Long	SDR	2B	RR	*
Subtract Normalized, Long	SD	6B	RX	*
Subtract Normalized, Short	SER	3B	RR	*
Subtract Normalized, Short	SE	7B	RX	*
Subtract Unnormalized, Long	SWR	2F	RR	*
Subtract Unnormalized, Long	SW	6F	RX	*
Subtract Unnormalized, Short	SUR	3F	RR	*
Subtract Unnormalized, Short	SU	7F	RX	*

أوامر المؤول

تعريف المعطيات	DC DS CCW	Assemblage conditionnel	تأويل مشروط	MACRO MNOTE MEXIT MEND
تقطيع	START CSECT DSECT COM ENTRY EXTRN			ACTR AGO AIF ANOP GBLA GBLB GBLC LCLA LCLB LCLC SETA SETB SETC
تعريف المراصف القاعدية	USING DROP			
مراقبة الذاكرة	TITLE EJECT SPACE PRINT			
مراقبة البرنامج	EQU ORG LTORG CNOP END COPY PUNCH REPRO ISEQ ICTL PUSH POP OPSYN			

مميزات الثوابت

النوع	الطول الضمعي	حدّ الإصطفااف	يتميّز بـ	بئر أو ملء إلى
III	-	بليّة	سهل	اليمن
N	-	بليّة	أرقلم ساحن عشرية	اليسار
II	-	بليّة	أرقلم ثنائية	اليسار
F	4	كلمة	أرقلم عشرية	اليسار
H	2	نصف كلمة	أرقلم عشرية	اليسار
E	4	كلمة	أرقلم عشرية	اليمن
D	8	كلمة مزدوجة	أرقلم عشرية	اليمن
L	16	كلمة مزدوجة	أرقلم عشرية	اليمن
I'	-	بليّة	أرقلم عشرية	اليسار
Z	-	بليّة	أرقلم عشرية	اليسار
A	4	كلمة	تعبير	اليسار
Y	2	نصف كلمة	تعبير	اليسار
G	2	نصف كلمة	تعبير	-
M	4	كلمة	مز قابل للتقل	اليسار

الكود الحرفي موسع

كود العملية الحرفي	للمعنى ...	التعليمة المولدة	الفتاع
B ... BR ... NOP ... NOPR ...	تفرع غير مشروط لا عملية	BC 15,... BCR 15,... BC 0,... BCR 0,...	1111 0000
BH ... BHR ... BL ... BLR ... BE ... BER ... BNH ... BNHR ... BNL ... BNLR ... BNE ... BNER بعد تعليقات المقارنة تفرع إذا كان : $Op(\bullet) \geq 2$ المتأثر > المتأثر 1 (\bullet) " < " " = " " ≤ " " ≥ " " = "	BC 2,... BCR 2,... BC 4,... BCR 4,... BC 8,... BCR 8,... BC 13,... BCR 13,... BC 11,... BCR 11,... BC 7,... BCR 7,...	0010 0100 1000 1101 1011 0111
BO ... BOR ... BP ... BPR ... BM ... BMR ... BNP ... BNPR ... BNM ... BNMR ... BNZ ... BNZR ... BZ ... BZR بعد التعليقات الحسابية تفرع إذا كانت النتيجة فيض عن السعة > 0 ... < 0 ... ≤ 0 ... ≥ 0 ... = 0 ... = 0	BC 1,... BCR 1,... BC 2,... BCR 2,... BC 4,... BCR 4,... BC 13,... BCR 13,... BC 11,... BCR 11,... BC 7,... BCR 7,... BC 8,... BCR 8,...	0001 0010 0100 1101 1011 0111 1000

(*) المقصود هما المتأثران 1 و 2 في تعليمة المقارنة .

ملاحظة : الكود الحرفي التذكيري المنتهي بحرف ٢ : يؤكد تعليمات من النسق RR . للمصنف المذكور يحتوي على عنوان التفرع .

مثلا : BR 3 : تفرع غير مشروط إلى العنوان الواقع في الموضع ٣ .

B ALPHA : تفرع غير مشروط إلى العنوان ALPHA .

ترجمة الملاحظات الواردة في بعض البرامج الموجودة في الكتاب

السطر	الملاحظة	الصفحة
5	ثوابت ميات . لا يوجد اصطفاك خاص . الطول 256	69
6	تأطير إلى اليسار . يتر إلى اليمين	
8	بتر إلى اليمين .	
9	تأطير إلى اليسار تكمله فراغات .	
10	توليد فاصلة عليا واحدة .	
11	نفس الملاحظة	
12	تكرار ووتر	
15	ثوابت مئاس عشرية . تأطير إلى اليمين . بتر إلى اليسار .	
16	طول ضمني	
17	طول ظاهري .	
18	بتر .	
21	ثوابت ثنائية . الطول الأقصى 256 بايت تأطير إلى اليمين .	
22	تكمله أصفار إلى اليسار . اصطفاك على البايت .	
23	ثنائي	
24	بتر إلى اليسار .	
25	نتر .	
26	تكرار .	
29	ثوابت بالفاصلة الثابتة على كلمة (F) أو نصف كلمة (H) .	
30	اصطفاك على الكلمة أو نصف الكلمة . عندما يكون الطول	
31	محددا لا يعود هناك اصطفاك . الثابتة هي بالنظام العشري	
34	إزاحة 3 بتات إلى اليسار (8 *)	
36	إزاحة 3 بتات إلى اليمين (/8)	
39	مدور أحل	
40	مدور أصغر .	
42	تعديل الطول LONG والاصطفاك ALIGN .	
44	إزاحة بتين إلى اليسار .	
49	ثوابت بالفاصلة المتحركة والدقة البسيطة . اصطفاك على الكلمة	70
50	تأطير إلى اليمين . لا بتر . القيمة مدورة .	
51	الطول الضمني 4 بايتات .	

- 52 بفاصلة متحركة
57 ثوابت بالفاصلة للتحركة وبالذقة للزودجة
58 اصطفايف عل الكلمة للزودجة . تأطير إلى اليمين . لا يتر
59 القيمة مدوّرة . الطول الضمفي 8 بايتات
66 ثوابت بالفاصلة للتحركة وبالذقة للزودجة
67 اصطفايف عل الكلمة للزودجة . الطول الضمفي 16 بايتة .
68 لا يتر . القيمة مدوّرة . -أس من 85- إلى 75+ .
73 ثوابت عشرية . الطول الأقصى يبلغ
74 16 بايتة . الإشارة تقع في الربع الأيسر
75 من البايته اليمنى الأخيرة . تأطير إلى اليمين . يتر إلى اليسار .
76 $X'F$ أو XC' في موقع الإشارة يُعتبران مثل +
77 $X'E'$ أو $X'D'$ في موقع الإشارة يُعتبران مثل -
78 لا يتم ترجمة الفاصلة العشرية أبداً إلى التالي .
79 تأطير إلى اليمين . يتر إلى اليسار .
80 الطول الضمفي .
82 يتر إلى اليسار
86 الثوابت العشرية للكثافة (Packed)
87 نفس قواعد الثوابت السابقة .
88 تقع الإشارة في الربع الأيمن الأخير .

- 2 رمز خارجي
6 رمز قابل للتقل
9 ثوابت عنوان من النوع A
10 تُكتب DC A (تعبير مطلق أو قابل للتقل)
11 اصطفايف عل الكلمة . الطول الضمفي 4 بايتات .
12 الأطوال الظاهرة للمكّنة هي من 1 إلى 4 بايتات .
13 يتر إلى اليسار . يمكن التحديد في كود حرفي .
18 طول ظاهر
20 رمز خارجي
23 ثوابت عنوان من النوع Y
24 تُكتب DC Y (تعبير مطلق أو قابل للتقل)
25 اصطفايف عل نصف الكلمة . الطول الضمفي: نصف كلمة .
26 الأطوال الظاهرة للمكّنة هي من 1 أو 2 بايتة .
27 يتر إلى اليسار . يمكن التحديد في كود حرفي .
29 لاحظوا أنّ للنتيجتين
30 تساويان $B + 2$
31 الطول الظاهر
32 يتر إلى اليسار
35 ثوابت عنوان من النوع S
36 تُكتب DC S (تعبير مطلق) .
37 أو DC S (تعبير قابل للتقل) .
38 أو DC S (تعبير مطلق (تعبير مطلق)).

السطر الملاحظة

الصفحة

73

39 مؤولة في نصف كلمة . مصطفة على نصف الكلمة .

40 لا يمكن تمثيلها في كود حرفي .

42 القاعدة (Base) 0 ، الإزاحة (Déplacement) = 1024

43 قاعدة وإزاحة RELOC

49 ثوابت عنوان من النوع V

50 تُستعمل فقط للعناوين الخارجية من النوع اسم البرنامج NOU-DE-PROG .

51 تكتب VDC (رمز خارجي قابل للنقل)

52 لا يرد الرمز القابل للنقل في أمر خارجي .

53 الطول الضمني 4 بليتات . معدل الطول = 3 أو 4

54 اصطفاك على حد كلمة ، بإمكانه أن يظهر في كود حرفي .

55 يؤخذ المؤول كلمة صفر .

79

3 متالية الدخول

4 و5 حفظ المرافف من شحن مرصف القاعدة

6 RI2 = مرصف القاعدة

7 البرنامج التالي

14 اصطفاك كلمة

18 (1) القاعدة 12 ظاهرة

19 (2) القاعدة 12 ظاهرة

21، 22 و23 كلّ التعاليمات من (3) حتى (7) تشحن X'88ABCDEF في المرصف 3 . الكتابة (3) هي الوحيدة

المستقلة عن مكان ALPHA بالنسبة إلى عنوان القاعدة .

24 و25 (3) استعمال تعبير قابل للنقل . قاعدة ضمنية .

26 (4) عملية تماثل رمزاً مطلقاً .

30 (7) استعمال كود حرفي

32 (8) "g" هي عبارة عن إزاحة

24 (9) خطاً اصطفاك

36 (10) "12" هي عبارة عن إزاحة

37 (11) "12" هي عبارة عن مرصف قاعلي

38 (12) خطاً في النحو

39 (13) خطاً في النحو

40 (14) خطاً في النحو

41 (15) 12 هي عبارة عن مؤشر

111

3 مؤشر (مصوب) إلى عنصر من TAB

4 مرصف إضافة إلى BXLE

5 مرصف مرجع إلى BXLE

6 مرصف عمل

11 القاعدة = المرصف 12

16 تصفير (إعداد)

23 طول الكلمة

111

- 47 حلقة مسح الجدول
48 و49 في حال عدم التبديل يتم فوز (ترتيب) TAB
52 تصنيف للمؤشر
53 تصنيف مرجع BXLE
56 عنصر إيسر في مرصف العمل
57 مقارنة
60 تبديل
62 تمثيل موقع INDIC
101 منطقة المعطيات
102 عدد عناصر TAB
105 إعداد INDIC

115

- 39 مؤشر بداية الجدول الثاني
40 مؤشر نهاية الجدول الثاني
41 مؤشر المنتصف والرتبة
42 مرصف العمل
43 طول العنصر
62 عدد عمليات التكرار في البرنامج
76 إعداد
80 حساب عنوان العنصر الوسط (للمنتصف)
84 قسمة على $L * 2$
85 (PTRELEM) = عدد العناصر في الجدول الثاني
87 إذا 0 فترجم حتى 1
89 ضرب بـ <
91 مقارنة
92 تفريع إذا كان ELEM < (MOT)
93 تفريع إذا كان ELEM > (MOT)
95 وجدنا العنصر حساب رتبة العنصر = (MOT)
98 قسمة على الطول
100 طباعة الرتبة والقيمة

116

- 129 لم نجده
153 منطقة المعطيات
154 عدد كلمات الجدول
155 طول العنصر

160

- 3 حفظ مرصيف للتادي
5 تعريف وشحن مرصيف القاعدة
6 تأخذ المرصيف 12 كقاعدة
7 عنوان PROGI في 12
8 حفظ R13 في المنطقة SAVE AREA من البرنامج .

الصفحة

10 و 11 حفظ عنوان المنطقة SAVE AREA من هنا البرنامج في المنطقة AREA من التالي 160

16 تعريف المنطقة SAVE AREA

22 متالية نداء PROGK

29 متالية العودة إلى PROGI

فهرست

الموضوع	الصفحة
تقديم	5
تمهيد	7

القسم الأول : عموميات

1- الآلة البسيطة	9
2- توكويد المعلومات	20
تمارين	35
3- العنوان المطلقة ، العنوان النسبية	36
4- هيكلية الحاسبات IBM 360/370	41
5- لغة الآلة	45
6- لغة المؤول	51

القسم الثاني 360/370

7- العناصر الأساسية	59
8- توجيهات تعريف الرموز	67
تمارين	75
9- كتابة العناوين بلغة المؤول	76
10- التعليمات بلغة المؤول ، عموميات	81
11- الحساب بفاصلة ثابتة والحركات	84
تمارين	92
12- التفريعات	93
تمارين	98

99	13 - العمليات المنطقية
104	تمارين
106	14 - عمليات الإزاحة
109	تمارين
110	15 - مسائل
117	16 - الحساب العشري
120	17 - الحساب بفاصلة متحركة
123	18 - تعليمات التحويل والتمثيل
127	تمارين
129	19 - الانقطاع والادخال والاخراج
138	20 - الأوامر المتعلقة بالعنونة وتركيبية المربّع
152	21 - البرامج الثانوية
162	22 - التأويل المشروط وماكرو التعليمات
177	23 - نصائح في البرمجة
181	حلول التمارين
189	ملحقات
190	جدول توكويد السمات
192	جدول أبجدي للتعليمات
195	أوامر المؤول
196	مميزات الثوابت
197	الكود الحرفي موسّع

هذا الكتاب

تعتبر لغة المؤؤل (الأسمبلر) من العناصر الأساسية في التفكير حول طريقة البرمجة بإحدى اللغات المتطورة فهي تتيح لنا فهماً مفصلاً لأليات الحاسب وليس بالإمكان الاستغناء عنها في إعداد المعلومات .

وتتجلى ضرورة استعمال لغة المؤؤل ، بالرغم من قوة اللغات المتطورة ، عندما يوجد إلزامات بالنسبة لفترات الإجابة (بعض البرامج الكبيرة ، أنظمة التشغيل ، المصروفات ، الوقت الحقيقي ، ...) أو بالنسبة لحجم الذاكرة (الحاسبات الصغيرة والمتوسطة) ، أو أيضاً لإلزامات تعود إلى عدم كفاية إمكانيات البرامج (فورتران ، باسك) .

من جهة أخرى ، سوف يجد مستعملو الميكرومعلوماتية في تطبيق لغة المؤؤل حلاً ممتازاً لما يعترضهم من مشاكل .

يتوجه هذا الكتاب إلى الطلاب والممارسين الذين يرغبون بتعميق معرفتهم في مجال المعلوماتية . وهو يتكون من فصول قصيرة ويتبدى انطلاقاً من ملاحظات بسيطة جداً على حاسبة الجيب ، بشكل يقود معه القارئ شيئاً فشيئاً ، لا سيما بفضل التمارين المحلولة والمفاهيم الأساسية في بنية الآلة ، إلى دراسة المؤؤل والماكرو - لغة . ولا شك أنه بالإمكان استعماله كمرجع ولتدريس متعلق بسلسلة الآلات المعتمدة كمثال (سلاسل 4000 ، 3000 ، IBM370) ولكنه وضع كي يكون دليلاً عاماً يوجه بطريقة سليمة أي برمجة بلغة المؤؤل .